

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DOTTORATO DI RICERCA IN

COMPUTER SCIENCE AND ENGINEERING

Ciclo 36

Settore Concorsuale: 09/H1 - SISTEMI DI ELABORAZIONE DELLE INFORMAZIONI

Settore Scientifico Disciplinare: ING-INF/05 - SISTEMI DI ELABORAZIONE DELLE INFORMAZIONI

ADVANCING ABSTRACTIVE LONG-INPUT SUMMARIZATION IN LOW-RESOURCE REGIMES: METHODS, DATASETS AND BENCHMARKS

Presentata da: Luca Ragazzi

Coordinatore Dottorato

Ilaria Bartolini

Supervisore

Gianluca Moro

Co-supervisore

Ilaria Bartolini

Esame finale anno 2024

Abstract

English

The field of natural language processing (NLP) has experienced remarkable growth in recent years, driven by the emergence of transformer-based models. In today's information-rich era, automatic text summarization has become a pivotal pursuit within NLP. It aims to distill complex textual documents into their essential elements, providing valuable support to experts engaged in labor-intensive tasks. Technically, extractive summarization identifies and extracts key sentences from the source text, whereas abstractive summarization amalgamates, rephrases, and paraphrases essential topics from the input to generate the synthesis. However, significant advances are often impeded by resource constraints. State-of-the-art solutions face challenges in low-resource regimes, relying on substantial computational power and extensive collections of annotated examples for effective training. This heavy dependence on resources poses a significant obstacle for small and medium-sized organizations, limiting their ability to manage these unsustainable costs. This thesis delves into the challenges of text summarization in low-resource environments, presenting multiple techniques categorized by methodology to advance abstractive summarization of long inputs. We show that text segmentation, retrieval-augmented generation, and knowledge injection and distillation are integral components of the solution, whose effectiveness is complemented by the availability of new rigorous datasets and benchmarks. Overall, this work represents a significant step towards real-world practical solutions that do not necessitate reliance on extensive manually curated data and high-memory GPUs.

Keywords – artificial intelligence, natural language processing, abstractive text summarization, transformers, large language models, low-resource regimes.

Italiano

Il campo dell'elaborazione del linguaggio naturale (NLP) ha registrato una crescita notevole negli ultimi anni, supportata dall'introduzione di nuovi modelli basati sul modello transformer. In un'epoca caratterizzata da un incessante flusso di informazioni, il compito di sintesi automatica del testo ha assunto un ruolo di particolare rilevanza all'interno del NLP. Questa attività mira a sviluppare sistemi in grado di generare sintesi concise e significative di documenti testuali, supportando gli esperti in compiti intensivi. Tecnicamente, la sintesi estrattiva identifica ed estrae frasi chiave dal testo originale, mentre la sintesi astrattiva aggrega e riformula i temi essenziali dell'input per generare il riassunto. Tuttavia, il suo successo è spesso oscurato dalla crescente richiesta di risorse computazionali. Le soluzioni all'avanguardia soffrono i regimi a basse risorse, poiché fanno affidamento su una potenza di calcolo sostanziale e su ampie collezioni di esempi annotati per un addestramento efficace. Questa forte dipendenza dalle risorse costituisce un ostacolo significativo per le organizzazioni di piccole e medie dimensioni, limitando la loro capacità di gestire questi costi insostenibili. Questa tesi approfondisce le sfide della sintesi del testo in ambienti a bassa risorsa, presentando diverse tecniche categorizzate per metodologia per avanzare la sintesi astrattiva di lunghi documenti. Mostriamo che la segmentazione del testo, la generazione con recupero e l'iniezione e distillazione della conoscenza sono componenti integrali della soluzione, la cui efficacia è integrata dalla disponibilità di nuovi dataset e benchmark rigorosi. Nel complesso, questo lavoro rappresenta un passo significativo verso soluzioni pratiche nel mondo reale che non richiedono una dipendenza da numerosi dati curati manualmente e da GPU con memoria elevata.

Parole chiave – intelligenza artificiale, elaborazione del linguaggio naturale, sintesi astrattiva del testo, transformers, modelli linguistici di grandi dimensioni, regimi a basse risorse.

"Con questo cosa voglio dire?
Non lo so. Però c'ho ragione e i fatti mi cosano."
– Palmiro Cangini

Acknowledgements

English

Throughout this three-year journey in academia, I realized that the pursuit of knowledge is both a thrilling adventure and an enigmatic mystery, where ideas dance like caffeinated squirrels. Undoubtedly, I express sincere gratitude to all those who contributed to the completion of this goal.

First of all, I thank my advisor, Prof. Gianluca Moro, whose support and guidance undeniably made all this and more possible. I learned to grow as a researcher noting all the possible minutiae that only experience can teach. I also thank Gruppo Maggioli for funding my Ph.D. scholarship.

Along the way, I had the pleasure of collaborating with my teammate Lorenzo (Lori). I enjoyed the long car conversations, the constant brainstorming of new ideas, and the nuanced complaints that popped like fireworks.
I thank Giacomo (Jack), not just a teammate, but my crazy and desperate sleepover buddy. Nu taca, the farm on the French Riviera is not yet our business (I see your point). Now, a new spritz cycle begins again and I am sure your answer will be: "oookaaayyy".
I extend my thanks to Stefano for joining me on my first journey overseas for a scientific paper presentation. I'm really hoping we can have another amazing experience together, just like that unforgettable week we had in Washington.
I also thank Paul for his light-heartedness and friendliness, and extend a warm welcome to the new additions, Cocco and Molfi for infusing "Bimbe di Naz" with a positive boost. Don't worry, I will introduce you to Giovanni La Penna too.

I thank my friends, particularly Delo, for providing unwavering certainty and being a reliable shoulder on which to rely whenever needed.
I sincerely thank my entire family for being there always.

Finally, a special acknowledgement is reserved for the person who not only opened up my path of teaching, but, above all, paved the way for a new life, together. Thank you, Veronika.

Italiano

Durante questo viaggio di tre anni nel mondo accademico, ho capito che la ricerca della conoscenza è sia un'avventura avvincente che un enigmatico mistero, dove le idee danzano come scoiattoli incaffeinati. Senza dubbio, provo una sincera gratitudine verso tutti coloro che hanno contribuito al raggiungimento di questo obiettivo.

Innanzitutto ringrazio il mio relatore, il Prof. Gianluca Moro, il cui sostegno e guida hanno indiscutibilmente reso possibile tutto questo e molto altro. Ho imparato a crescere come ricercatore annotando ogni possibile minuzia che solo l'esperienza può insegnare. Ringrazio inoltre Gruppo Maggioli per aver finanziato la mia borsa di dottorato.

Lungo il percorso ho avuto il piacere di collaborare con il mio compagno di squadra Lorenzo (Lori). Mi sono piaciute le lunghe conversazioni in macchina, il costante brainstorming di nuove idee e le lamentele sfumate che scoppiettavano come fuochi d'artificio.
Ringrazio Giacomo (Jack), non solo un compagno di squadra, ma il mio pazzo e disperatissimo compagno di pigiama party. Nu taca, la fattoria in Costa Azzurra non è ancora affare nostro (I see your point). Ora ricomincia un nuovo ciclo di spritz, e sono sicuro che la tua risposta sarà: "oookaaayyy".
Ringrazio Stefano per avermi accompagnato nel mio primo viaggio all'estero per la presentazione di un articolo scientifico. Spero davvero di poter fare un'altra esperienza insieme, proprio come quella settimana indimenticabile trascorsa a Washington.
Ringrazio anche Paul per la sua spensieratezza e simpatia, e do un caloroso benvenuto ai nuovi arrivati, Cocco e Molfi, per aver infuso una spinta positiva nel gruppo "Le Bimbe di Naz". Non preoccupatevi, vi presenterò anche a voi Giovanni La Penna.

Ringrazio i miei amici, in particolare Delo, per fornirmi una certezza incrollabile e per essere stato una spalla affidabile su cui appoggiarsi quando necessario.
Ringrazio di cuore tutta la mia famiglia per essere sempre presente.

Infine, un riconoscimento speciale è riservato alla persona che non solo ha aperto il mio percorso di insegnamento ma, soprattutto, ha aperto la strada a una nuova vita, insieme. Grazie Veronika.

Contents

Abstract i

1 Introduction 1
- 1.1 Natural Language Processing . . . 1
- 1.2 Abstractive Long-Input Summarization . . . 3
- 1.3 Low-Resource Regimes . . . 5

2 Research Overview 7
- 2.1 Text Segmentation . . . 8
- 2.2 Retrieval-Augmented Generation . . . 9
- 2.3 Knowledge Injection and Distillation . . . 10
- 2.4 Datasets and Benchmarks . . . 12

3 Text Segmentation 14
- 3.1 Semantic Self-Segmentation for Abstractive Summarization of Long Documents in Low-Resource Regimes . . . 14
- 3.2 Align-Then-Abstract Representation Learning for Low-Resource Summarization . . . 29
- 3.3 Efficient Memory-Enhanced Transformer for Long-Document Summarization in Low-Resource Regimes . . . 43

4 Retrieval-Augmented Generation 59
- 4.1 Discriminative Marginalized Probabilistic Neural Method for Multi-Document Summarization of Medical Literature . . . 59
- 4.2 Retrieve-and-Rank End-to-End Summarization of Biomedical Studies . . . 71

5 Knowledge Injection and Distillation 82
- 5.1 Graph-based Abstractive Summarization of Extracted Essential Knowledge for Low-Resource Scenarios . . . 82
- 5.2 REVELIO: Interpretable Long-Form Question Answering . . . 96
- 5.3 Enhancing Legal Question Answering with Data Generation and Knowledge Distillation from Large Language Models . . . 104
- 5.4 Un*kn*own Claims: Generation of Fact-Checking Training Examples from Unstructured and Structured Data . . . 121

6 Datasets and Benchmarks **138**
6.1 Carburacy: Summarization Models Tuning and Comparison in Eco-Sustainable Regimes with a Novel Carbon-Aware Accuracy . . . 138
6.2 Abstractive Summarization Through the Prism of Decoding Strategies . . 152
6.3 Multi-Language Transfer Learning for Low-Resource Legal Case Summarization . . . 192
6.4 COMMA: A Multi-Task and Multi-Lingual Dataset of Constitutional Verdicts . . . 214

7 Conclusion and Future Work **254**
7.1 Discussion . . . 254
7.2 Visionary Directions . . . 256

Chapter 1

Introduction

In the vast landscape of information that permeates our digital world, the ability to distill, comprehend, and articulate the core of textual content is of paramount importance. Automatic text summarization [Khan et al., 2023] is the cornerstone of natural language processing (NLP) and plays a key role as a facilitator in the navigation of this sea of information. It concerns the development of algorithms to extract essential elements from documents and convey a concise version, e.g., an abstract of a scientific paper. As we stand on the precipice of an information age marked by exponential data proliferation, the position of text summarization becomes increasingly critical, influencing how we access, understand, and disseminate knowledge. Furthermore, in a world where time is a precious commodity and attention spans are transitory, summarization serves as a bridge between data and the human need for efficient comprehension, offering tools to empower decision-making, research, and communication.

1.1 Natural Language Processing

Natural language processing (NLP) is a prominent subfield within the expansive domain of artificial intelligence (AI), dedicated to the nuanced interaction between computers and human languages. At the convergence of computer science, machine learning, and computational linguistics, NLP emerges as the main subject to automate the analysis, comprehension, and generation of human language content [Min et al., 2024].

The scope of NLP is extensive and encompasses a wide range of tasks, including speech recognition, language understanding, sentiment analysis, machine translation, and text summarization. The trajectory of NLP evolution has been markedly influenced by the development of robust deep learning techniques, such as recurrent neural networks (RNNs) [Sherstinsky, 2018], convolutional neural networks (CNNs) [Gu et al., 2018], and, more recently, transformer-based models [Vaswani et al., 2017]. These advances have propelled the field into new realms of capability, enabling machines to process and generate human-like language with unprecedented accuracy and efficiency [Zhou et al., 2023b]. Of particular note are the transformative changes observed in the hardware infrastructure, which have led to innovative breakthroughs in model architecture that significantly

increase the capabilities of text generation models [Ignat et al., 2023]. The influence of these advances resonates prominently within the domain of text summarization, offering novel avenues to enhance the efficiency and effectiveness of algorithms.

1.1.1 Transformers and Pretrained Language Models

The transformer is one of the pivotal architectural advances in NLP. Exemplified by models such as encoder-only Bert [Devlin et al., 2019] and decoder-only Gpt [Radford et al., 2019a], these networks have demonstrated remarkable prowess in capturing contextual information and semantic nuances. In the context of summarization, transformers facilitate a more subtle understanding of the relationships between words, phrases, and sentences, promoting a departure from conventional extractive summarization methods [Yadav et al., 2021]. In fact, unlike extraction-based approaches, transformers enable the generation of abstractive summaries, namely, to produce new contextually relevant sentences that encapsulate the essential meaning of the source text. This shift towards generative summarization represents a leap forward in designing models to have a deeper level of comprehension and context-awareness and to shorten the information in a way that aligns more closely with human-like summarization processes [Yang et al., 2019].

The advent of pretrained language models (PLMs) has revolutionized NLP [Qiu et al., 2020], allowing models to learn from large general-purpose datasets before being fine-tuned for downstream-specific tasks, such as summarization or question-answering. The purpose of pretraining is to equip models with a broad understanding of linguistic structures, semantics, and contextual relationships through unsupervised learning strategies such as masked language modeling [Sinha et al., 2021] or autoregressive language modeling [Radford et al., 2019a]. Pioneering encoder-decoder PLMs, such as Bart [Lewis et al., 2020a] and T5 [Raffel et al., 2020], show unprecedented language generation capabilities in complex tasks such as summarization [Liu and Lapata, 2019b]. Recently, even decoder-only alternatives proved to be valuable summarizers [Wang et al., 2023].

1.1.2 The Era of Large Language Models

Following advances in model architectures, particularly with the widespread success of transformers and new hardware capabilities, we are witnessing a new NLP era distinguished by large language models (LLMs). Specifically, this epoch is characterized by the development and deployment of language models of unprecedented scale and complexity, prominently exemplified by models such as Gpt-3 with 175 billion parameters [Brown et al., 2020]. This monumental scale empowers models with an unparalleled ability to capture intricate linguistic patterns and generate human-like text across a spectrum of tasks [Zhao et al., 2023].

The origin of open-source LLMs such as LLaMa [Touvron et al., 2023a] has undeniably become a cornerstone in the promotion and advancement of research within NLP. The public availability of these models due to widely utilized frameworks, such as HuggingFace Transformers [Wolf et al., 2019],[1] has democratized access to cutting-edge

[1] https://huggingface.co/docs/transformers/

language models, allowing researchers and practitioners around the world to contribute to the state-of-the-art of NLP.

Recently, a notable direction has emerged with the exploration of smaller-scale variants, exemplified by LLMs with 7B parameters. In fact, while large-scale LLMs with huge parameters have garnered considerable attention [OpenAI, 2023], there is a growing recognition of the impact of more compact counterparts that aim to balance computational efficiency with meaningful language understanding [Jiang et al., 2023, Tunstall et al., 2023]. Despite their reduced scale (i.e., ≈99% fewer parameters compared to PaLM-540B [Chowdhery et al., 2023]), these models exhibit remarkable capacity [Kim et al., 2023], opening avenues for applications in scenarios where high-budget computational resources (i.e., GPU cards with high dedicated RAM) are limited.

Limitations Although remarkable advances have been made, LLMs face several challenges that pose impediments to their reliable deployment in real-world scenarios. The primary concern revolves around the generation of hallucinations [Zhang et al., 2023], namely the fabrication of non-authentic information, raising doubts about the trustworthiness of the content they produce. This issue is also related to the inherent black-box nature of these models, which adds another layer of complexity, making it unfeasible to interpret and understand their decision-making processes. Current research efforts push retrieval-augmented generation (RAG) [Lewis et al., 2020b] and knowledge graphs (KGs) as one of the main techniques to alleviate hallucinations in LLMs [Tonmoy et al., 2024, Wu et al., 2024] and enhance interpretability [Pan et al., 2023]. The second important problem concerns the difficulty in accurately processing long documents [Chang et al., 2023], which also requires substantial hardware computational power that affects the applicability of LLMs in commodity equipment.

1.2 Abstractive Long-Input Summarization

In this thesis, we deal with the problem of abstractive long-input summarization, which transcends conventional extractive methods by not only selecting and returning salient information, but also generating novel contextually-relevant content from long sources. Although the manifestation of hallucination problems in contemporary sequence-to-sequence models is apparent [Kryscinski et al., 2020, Rehman et al., 2023], abstractive syntheses give these models an increased level of creativity and linguistic fluency. In particular, when dealing with long documents characterized by specific details scattered throughout the text [Huang et al., 2021], generative models are proficient at linking and consolidating information from different parts of the document, ensuring a more comprehensive and informative summary. Overall, abstractive summarization exhibits a closer alignment with human-like comprehension and summarization, advancing the development of more natural and interpretable summaries to increase productivity in real-world applications.

In an epoch defined by information overload, the significance of proficient text summarization aligns seamlessly with the efficiency and adaptability of our knowledge-driven

society. Summarization applications are far-reaching and cover multiple domains, e.g., news articles [Goyal et al., 2022a], academic papers [Altmami and Menai, 2022], legal case documents [Jain et al., 2021b], and medical reports [Jain et al., 2022]. In business scenarios, it serves as a fundamental element in supporting information retrieval systems, contributing to the advancement of the convenience of conversational agents [Manathunga and Illangasekara, 2023] such as ChatGPT [OpenAI, 2023].

1.2.1 Practical Realities

In the pragmatic context of information consumption, the task of long-input summarization [Phang et al., 2023] emerges as a critical necessity, overshadowing the more commonplace focus on shorter texts.[2] Long documents inherently carry a wealth of information. As the volume of data increases, the human capacity to assimilate and comprehend every detail diminishes. For this reason, summarizing lengthy documents becomes an instrumental strategy in reducing information overload and promoting time-efficient decision-making.

Applications Long-input summarization encapsulates a spectrum of challenging tasks, each tailored to address distinct aspects of handling extended textual sources. A prominent facet is long document summarization (LDS) [Koh et al., 2023a], which specifically targets the condensation of comprehensive information present in a single long document. This task is particularly crucial in domains such as legal, biomedical, and academic research, where the assimilation of extensive documents, such as scientific papers [Gidiotis and Tsoumakas, 2020] or lawsuits [Shen et al., 2022], requires a rigorous approach to distill essential insights. On the other hand, multi-document summarization (MDS) [Ihsan et al., 2023] extends the scope of summarization tasks to involve the synthesis of information in multiple documents. MDS is paramount in scenarios where comprehensive knowledge extraction requires the amalgamation of concepts from various sources, e.g., the creation of systematic literature reviews [Liu et al., 2022a].

Challenges Both LDS and MDS represent critical real-world dimensions of text summarization. For these reasons, they have gained considerable momentum in current NLP research. As the length and complexity of textual sources amplify, conventional methods tailored for shorter texts encounter inherent limitations. Notably, standard transformers are limited in their capacity to process texts beyond 1024 tokens, necessitating the truncation of exceeding portions and overlooking potentially pertinent information. This truncation approach, especially in syntheses that encapsulate distributed information from the source text, is an ineffective strategy that results in a suboptimal conveyance of comprehensive summaries. Moreover, these solutions rely on time and memory complexity that scales quadratically with input length, leading to notably inefficient pipelines

[2]While there is no formal demarcation between short and long texts, it is widely acknowledged that texts containing fewer than 1024 tokens are generally considered short. This threshold aligns with the maximum input size accommodated by transformers like Bart [Lewis et al., 2020a].

when applied to extremely long texts. The recognition of such limitations has fueled an intensified focus on advancing methodologies and models tailored to address the challenges posed by articulated textual input. The prominent methodologies encompass hierarchical models [Li et al., 2021a], extract-then-abstract frameworks [Li et al., 2021b], and linear transformers [Tay et al., 2023].[3]

Related Tasks Several tasks align closely with the objectives of summarization and can act as valuable components within a summarization pipeline: (i) sentence paraphrasing [Zhou and Bhat, 2021] involves transforming an input sentence while preserving its original meaning, serving as a versatile tool to generate alternative expressions for data augmentation pipelines [Abaskohi et al., 2023]; (ii) sentence compression [Malireddy et al., 2020] is similar to paraphrasing but requires the output to be shorter; (iii) text simplification [Al-Thanyyan and Azmi, 2022] focuses on modifying an input sentence to enhance readability and comprehension. These text rewriting tasks play a pivotal role in the composition of lay summaries [Goldsack et al., 2022], facilitating the translation of intricate technical information into a more accessible format for a wider audience.

Summary Evaluation Accurate evaluation is essential to gauge the effectiveness of generative models and their suitability for real-world applications. In the absence of robust metrics, the assessment of inferred summaries becomes subjective and lacks foundation. Currently, manual evaluation, employing human annotators, stands as the most rigorous approach, directly capturing the expectations of real-life users. However, this method is notably time-consuming and expensive, prompting the development of automatic metrics that seek to approximate human judgments. The summary assessment encompasses multiple dimensions, from syntactic and semantic quality to factuality and redundancy. Nevertheless, the cornerstone of the evaluation metrics is ROUGE [Lin, 2004], which focuses only on lexical superficiality counting n-gram matches.

1.3 Low-Resource Regimes

In the landscape of NLP, the challenges posed by resource availability vary significantly, exerting a profound influence on the trajectory of research and its potential progress and applicability. At the one end of the spectrum, low-resource regimes represent real-world scenarios where constraints such as limited computational power and data availability prevail. In these practical settings, developing effective text summarizers requires innovative approaches that can operate efficiently within such constraints. In contrast, high-budget environments represent the opposite extreme, where big companies (e.g., OpenAI, Meta, Google) can afford ample financial resources and computational power to push the boundaries of model complexity and develop new pioneering LLMs.

This duality in resource availability underscores the need for text summarization methodologies that can adapt to both extremes, i.e., state-of-the-art eco-sustainable

[3]Further elaboration and detailed information on these approaches can be found in the subsequent chapters of this thesis.

models. As noted previously, addressing the challenges of low-resource regimes requires resource-efficient algorithms and models, whereas high-budget environments permit the exploration of cutting-edge techniques with the promise of unparalleled performance. The intersection of these two realms encapsulates the dynamic landscape in which text summarization solutions must flourish, catering to the diverse practicalities encountered in real-world applications.

Data & Hardware Scarcity Our research initiation in the field of automatic text summarization focused on the identification of two crucial challenges prevalent in current real-world research. First, generative models are highly dependent on large datasets for effective learning in synthesizing textual content, even when employing summarization-specific pretraining objectives [Zhang et al., 2020a, Xiao et al., 2022]. Second, the pivotal role of GPU hardware becomes apparent, especially when accommodating larger models whose performance demonstrates improvement with an increase in the number of parameters [Guo et al., 2022a]. These challenges become more pronounced when dealing with lengthy documents, primarily due to two key reasons: (i) summary-worthy information tends to be more dispersed within extensive texts [Huang et al., 2021], necessitating models to adeptly discern noise and pinpoint salient content during training; (ii) the demand for GPU memory escalates with the size of the input. Moreover, in a business scenario, incorporating long texts into a text summarization pipeline is a logical strategy, enabling users to improve their productivity by efficiently analyzing a large volume of documents and identifying the most relevant content for their needs.

Literature Overview In this delicate real context, prior works primarily emphasize the enhancement of performance by allowing models to consume more content. To achieve this goal, the introduction of sparse attention mechanisms, aimed at approximating the behavior of full self-attention in vanilla transformers [Vaswani et al., 2017], has proven to be crucial in outperforming quadratic models. Indeed, such solutions exhibit linear time and memory complexity in the input size, making them well-suited for processing long documents. Consequently, they are particularly advantageous in long-input summarization tasks [Beltagy et al., 2020, Huang et al., 2021, Guo et al., 2022a, Phang et al., 2023]. However, despite the performance enhancements and improved computational efficiency achieved by such methods, it is noteworthy that these approaches come with the following limitations: (i) they still require extensive training datasets for convergence and high-budget GPUs (e.g., 48 GB memory [Beltagy et al., 2020]); (ii) they still necessitate truncating very long inputs (i.e., >16,384 tokens [Guo et al., 2022a]) such as books [Wu et al., 2021a, Kryscinski et al., 2022]. For these reasons, our focus shifts to entirely different approaches to address the aforementioned issues.

Chapter 2

Research Overview

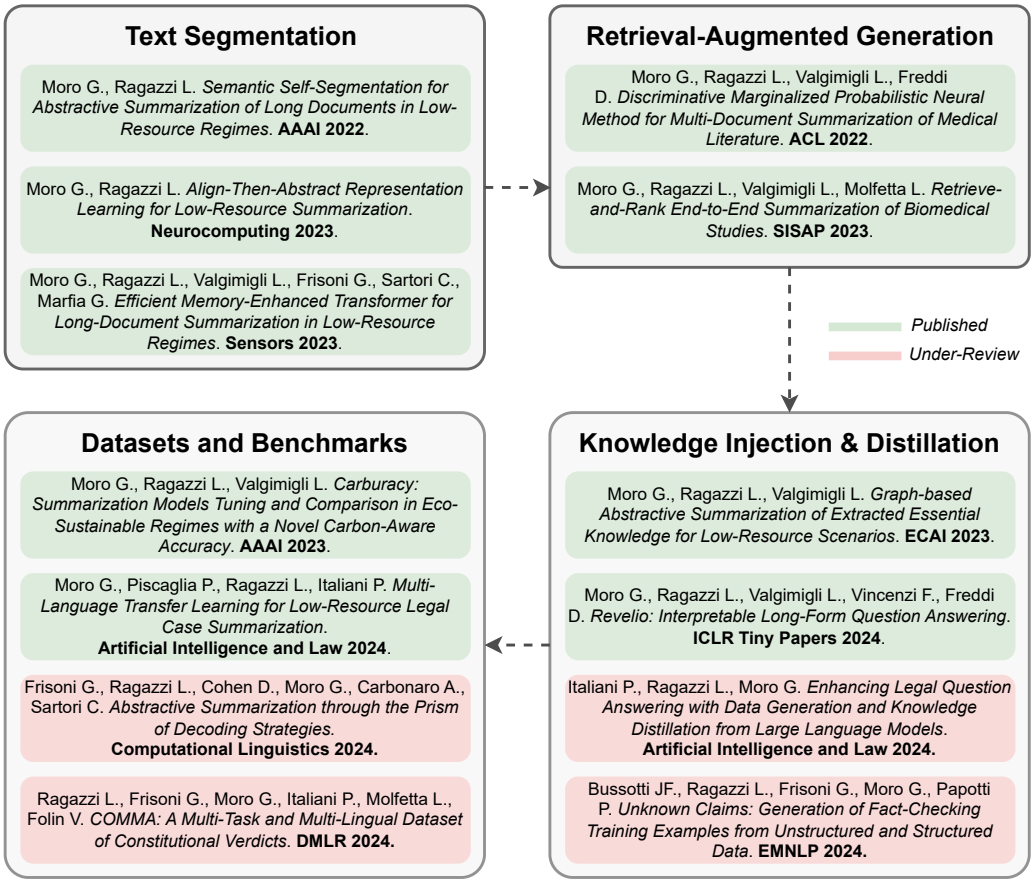


FIGURE 2.1. Holistic view of the research illustrating the connection between the works of this thesis.

This thesis is a compilation of several contributions, comprising nine open-access publications and four works currently under review. The organization of these papers is methodologically structured to address the challenges inherent in abstractive long-input summarization within low-resource regimes. It is imperative to emphasize that the content of the publications is reproduced without alterations to the original text, ensuring the preservation of its authenticity and integrity.[1]

The methodologies employed in these works encompass three key techniques: text segmentation, retrieval-augmented generation, and knowledge injection and distillation. Each of these approaches is instrumental in overcoming the unique obstacles posed by low-resource environments. Finally, our contribution extends to the provision of novel datasets and benchmarks, serving as foundational elements to stimulate research and facilitate the development of new, efficient, and real-world state-of-the-art solutions.

The subsequent sections introduce the motivation behind each method, elucidating their respective contributions to the overarching goals of this thesis.

2.1 Text Segmentation

2.1.1 Background

Text segmentation constitutes a fundamental task in NLP that involves the dissection of a continuous text stream into coherent and meaningful segments [Ghinassi et al., 2023]. These segments may manifest at various hierarchical levels, ranging from individual sentences and paragraphs [Aumiller et al., 2021] to more refined units, such as clauses or phrases [Lukasik et al., 2020]. The principal aim of text segmentation lies in enhancing the structural comprehension of a document, thereby expediting subsequent analytical processes and applications. In text summarization, this methodology has been used to partition long scientific documents into predefined sections. Subsequently, a summary is generated by concatenating the summaries of the corresponding sections [Gidiotis and Tsoumakas, 2020]. This approach is constrained by the following limitations: (i) its applicability is restricted to documents that are inherently structured into sections, such as academic papers; (ii) the chunk size is not determined based on the available hardware; (iii) the chunks are processed independently, leading to potential loss of context.

2.1.2 Research Contributions

Taking into account the practical challenges of low resources, in Chapter 3, we emphasize the importance of the text segmentation technique in addressing the foundational issues mentioned above. In particular, in Section 3.1, we demonstrate how text segmentation allows models to process extended inputs in a chunk-by-chunk manner, effectively resolving both memory and data scarcity issues. Furthermore, Section 3.2 illustrates how a segmentation solution can be trained collaboratively with a generative language

[1] The papers currently under review will be finalized in the thesis camera-ready version, incorporating a series of ongoing improvements.

model to acquire optimal document segmentation, thus facilitating the generation of the most accurate summary. Lastly, in Section 3.3, we introduce a mechanism that provides models with memory to retain relevant information from each segment, addressing the problem of context lost during independent chunk processing.

Publications

1. *Section 3.1*: Gianluca Moro and Luca Ragazzi. Semantic self-segmentation for abstractive summarization of long documents in low-resource regimes. In Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022, pages 11085–11093. AAAI Press, 2022. doi: 10.1609/AAAI.V36I10.21357. URL https://doi.org/10.1609/aaai.v36i10.21357

2. *Section 3.2*: Gianluca Moro and Luca Ragazzi. Align-then-abstract representation learning for low-resource summarization. Neurocomputing, 548:126356, 2023. doi: 10.1016/J.NEUCOM.2023.126356. URL https://doi.org/10.1016/j.neucom.2023.126356

3. *Section 3.3*: Gianluca Moro, Luca Ragazzi, Lorenzo Valgimigli, Giacomo Frisoni, Claudio Sartori, and Gustavo Marfia. Efficient memory-enhanced transformer for long-document summarization in low-resource regimes. Sensors, 23(7):3542, 2023d. doi: 10.3390/S23073542. URL https://doi.org/10.3390/s23073542

2.2 Retrieval-Augmented Generation

2.2.1 Background

Generative language models grapple with challenges stemming from a restricted context window, exemplified by the constraint of processing only 1024 [Lewis et al., 2020a] or 4096 input tokens [Beltagy et al., 2020] within the network, and the issue of hallucination, where the model may generate non-real information [Ji et al., 2023]. Retrieval-augmented generation (RAG) technique [Lewis et al., 2020b] represents an emerging paradigm in NLP where the content creation process involves a synergistic combination of retrieval and generative techniques. Unlike conventional sequence-to-sequence text generation models that rely solely on learned patterns, RAG integrates external knowledge sources to enrich the relevance, factual accuracy, and contextuality of the generated text. Previous solutions have focused exclusively on open-book question answering [Pan et al., 2022, Siriwardhana et al., 2023], overlooking the potential applicability of this approach to abstractive long-input summarization.

2.2.2 Research Contributions

In tasks where an input query is provided, e.g., query-focused multi-document summarization [Roy and Kundu, 2024], it is crucial to give to the model only the relevant information to reduce noise, as opposed to giving all document chunks. In Chapter 4, we show how the RAG-like technique [Lewis et al., 2020b] is key in furnishing models with only the relevant information that fits the model context window. Section 4.1 shows that, when combined with token marginalization at decoding time, we can effectively aggregate multiple sources, facilitating a cost-effective computation to discern the final summary. Subsequently, in Section 4.2, we demonstrate improved performance when the retrieval and generation modules are jointly trained in an end-to-end fashion.

Publications

1. *Section 4.1*: Gianluca Moro, Luca Ragazzi, Lorenzo Valgimigli, and Davide Freddi. Discriminative marginalized probabilistic neural method for multi-document summarization of medical literature. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 180–189. Association for Computational Linguistics, 2022a. doi: 10.18653/V1/2022.ACL-LONG.15. URL https://doi.org/10.18653/v1/2022.acl-long.15

2. *Section 4.2*: Gianluca Moro, Luca Ragazzi, Lorenzo Valgimigli, and Lorenzo Molfetta. Retrieve-and-rank end-to-end summarization of biomedical studies. In Oscar Pedreira and Vladimir Estivill-Castro, editors, Similarity Search and Applications - 16th International Conference, SISAP 2023, A Coruña, Spain, October 9-11, 2023, Proceedings, volume 14289 of Lecture Notes in Computer Science, pages 64–78. Springer, 2023e. doi: 10.1007/978-3-031-46994-7_6. URL https://doi.org/10.1007/978-3-031-46994-7_6

2.3 Knowledge Injection and Distillation

2.3.1 Background

External knowledge, typically in textual format, serves to guide models in focusing on specific information to address a given task, as exemplified in the RAG pipeline. However, the inclusion of heterogeneous sources, such as graphs or knowledge bases, can be crucial to providing additional information and improving task-solving capabilities [Hu et al., 2022b, Frisoni et al., 2023]. This procedure in NLP is known as knowledge injection [Hu et al., 2022a] and aims to supplement the inherent knowledge of models with factual knowledge, helping models adapt to specific domains [Emelin et al., 2022]. As an alternative strategy, considering the prevalence of LLMs [Zhao et al., 2023], it is recognized that these expansive networks inherently store a wealth of knowledge in their

parameters. Hence, knowledge distillation [Gou et al., 2021] involves transferring knowledge from a larger teacher model to a smaller student language model (SLM), producing cost-efficient solutions that remain competitive. Various distillation strategies include training SLMs either on data generated by LLMs [Zhou and Chiam, 2023] or directly on the probability distribution of their predictions [Stanton et al., 2021].

2.3.2 Research Contributions

Motivated by these considerations, Chapter 5 explores how knowledge can be injected into generative language models. We show that this technique is crucial for learning to select only the salient sentences from the source to reduce noise (Section 5.1) or to provide reasoning paths as rationales for a given model answer (Section 5.2). Furthermore, in Section 5.3, we illustrate how LLM knowledge distillation facilitates the generation of high-quality synthetic examples. Such artificial instances play a crucial role in overcoming the scarcity of publicly available data, enabling SLM training capable of surpassing LLM performance in legal question answering while remaining economically viable for deployment in low-resource environments. In Section 5.4, we demonstrate that even SLMs are capable of creating human-level training data by employing summarization strategies from given evidence to assist the training of fact-checking applications.

Publications

1. *Section 5.1*: Gianluca Moro, Luca Ragazzi, and Lorenzo Valgimigli. Graph-based abstractive summarization of extracted essential knowledge for low-resource scenarios. In Kobi Gal, Ann Nowé, Grzegorz J. Nalepa, Roy Fairstein, and Roxana Radulescu, editors, ECAI 2023 - 26th European Conference on Artificial Intelligence, September 30 - October 4, 2023, Kraków, Poland - Including 12th Conference on Prestigious Applications of Intelligent Systems (PAIS 2023), volume 372 of Frontiers in Artificial Intelligence and Applications, pages 1747–1754. IOS Press, 2023c. doi: 10.3233/FAIA230460. URL https://doi.org/10.3233/FAIA230460

2. *Section 5.2*: Gianluca Moro, Luca Ragazzi, Lorenzo Valgimigli, Fabian Vincenzi, and Davide Freddi. Revelio: Interpretable long-form question answering. In The Second Tiny Papers Track at ICLR 2024, Tiny Papers @ ICLR 2024, Vienna, Austria, May 7-11, 2024. OpenReview.net, 2024

Under Review

1. *Section 5.3*: Paolo Italiani, Luca Ragazzi, and Gianluca Moro. Enhancing legal question answering with data generation and knowledge distillation from large language models, 2024. Submitted to Artificial Intelligence and Law

2. *Section 5.4*: Jean Flavien Bussotti, Luca Ragazzi, Giacomo Frisoni, Gianluca Moro, and Paolo Papotti. Un*kn*own claims: Generation of fact-checking training examples from unstructured and structured data, 2024. Submitted to the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024

2.4 Datasets and Benchmarks

2.4.1 Background

State-of-the-art models and techniques achieve their prowess through large-scale resources and are rigorously evaluated on comprehensive testbeds (e.g., SQuAD [Rajpurkar et al., 2016], GLUE [Wang et al., 2019b], SuperGLUE [Wang et al., 2019a]). In practical scenarios, organizations commonly employ cutting-edge PLMs as a foundational framework to fulfill their downstream tasks [Abdelnabi et al., 2023]. However, there is a notable scarcity of studies in the literature that provide model benchmarks considering their eco-sustainability, an aspect highly relevant in real-world applications. A primary goal of product teams is to have competitive models at affordable costs and a clear guide to the utilization of such solutions, including the determination of optimal hyperparameters based on the use case.

2.4.2 Research Contributions

In response to these practical needs, Chapter 6 presents valuable benchmarks that specifically focus on the utilization of energy-saving models with the proposal of the first carbon-aware accuracy metric (Section 6.1) and a massive fine-grained study on decoding-time strategies for text summarizatin (Section 6.2). Additionally, to make a meaningful contribution to research in a socially impactful field, we present a multilingual legal benchmark, underscoring the importance of transfer learning to overcome data scarcity in specific jurisdictions (Section 6.3). Moreover, we introduce a novel legal dataset covering multiple valuable tasks and languages, drawn from documents of the Constitutional Court of the Italian Republic (Section 6.4).

Publications

1. *Section 6.1*: Gianluca Moro, Luca Ragazzi, and Lorenzo Valgimigli. Carburacy: Summarization models tuning and comparison in eco-sustainable regimes with a novel carbon-aware accuracy. In Brian Williams, Yiling Chen, and Jennifer Neville, editors, Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023, pages 14417–14425. AAAI Press, 2023b. doi: 10.1609/AAAI.V37I12.26686. URL https://doi.org/10.1609/aaai.v37i12.26686

2. *Section 6.3*: Gianluca Moro, Nicola Piscaglia, Luca Ragazzi, and Paolo Italiani. Multi-language transfer learning for low-resource legal case summarization. Artificial Intelligence and Law, pages 1–29, 2023a. doi: 10.1007/s10506-023-09373-8. URL https://doi.org/10.1007/s10506-023-09373-8

Under Review

1. *Section 6.2*: Giacomo Frisoni, Luca Ragazzi, David Cohen, Gianluca Moro, Antonella Carbonaro, and Claudio Sartori. Abstractive summarization through the PRISM of decoding strategies, 2024. Submitted to Computational Linguistics

2. *Section 6.4*: Luca Ragazzi, Giacomo Frisoni, Gianluca Moro, Paolo Italiani, Lorenzo Molfetta, and Veronika Folin. COMMA: A multi-task and multi-lingual dataset of constitutional verdicts, 2024. Submitted to Data-centric Machine Learning Research

Chapter 3

Text Segmentation

3.1
Semantic Self-Segmentation for Abstractive Summarization of Long Documents in Low-Resource Regimes

Originally published as: *Gianluca Moro and Luca Ragazzi. Semantic self-segmentation for abstractive summarization of long documents in low-resource regimes. In Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022, pages 11085–11093. AAAI Press, 2022. doi: 10.1609/AAAI.V36I10.21357. URL https://doi.org/10.1609/aaai.v36i10.21357*

Abstract

The quadratic memory complexity of transformers prevents long document summarization in low computational resource scenarios. State-of-the-art models need to apply input truncation, thus discarding and ignoring potential summary-relevant contents, leading to a performance drop. Furthermore, this loss is generally destructive for semantic text analytics in high-impact domains such as the legal one. In this paper, we propose a novel semantic self-segmentation (SE3) approach for long document summarization to address the critical problems of low-resource regimes, namely to process inputs longer than the GPU memory capacity and produce accurate summaries despite the availability of only a few dozens of training instances. SE3 segments a long input into semantically coherent chunks, allowing transformers to summarize very long documents without truncation by summarizing each chunk and concatenating the results. Experimental outcomes show the approach significantly improves the performance of abstractive summarization transformers, even with just a dozen of labeled data, achieving new state-of-the-art results on two legal datasets of different domains and contents. Finally, we report ablation studies

to evaluate each contribution of the components of our method to the performance gain.

3.1.1 Introduction

State-of-the-art solutions on abstractive summarization are built upon the transformer model [Vaswani et al., 2017] with quadratic time and memory complexities in the input size [Lewis et al., 2020a, Qi et al., 2020, Raffel et al., 2020, Zhang et al., 2020a]. Such models have been trained with short inputs, so they struggle to model long sequences accurately in downstream tasks. Thus, efficient transformers with linear complexity have been proposed to process longer sequences by reducing the attention mechanism calculation [Beltagy et al., 2020, Kitaev et al., 2020, Zaheer et al., 2020, Choromanski et al., 2021, Huang et al., 2021, Xiong et al., 2021, Guo et al., 2022a]. However, training large transformers requires high-resource settings [Ahmed and Wahed, 2020, Sharir et al., 2020], leaving long document summarization an open research problem in low-resource regimes with limited GPU memories and only dozens of labeled training instances.

One of the domains most affected by long documents and low-resource settings of labeled data is the legal one, where reading and evaluating legal cases are labor-intensive and time-consuming tasks for legal experts [Kornilova and Eidelman, 2019]. Legal texts are generally long with a complex and articulated structure, characterized by longer sentences than other domains that make up long reasonings, understandable only after reading the entire textual details [Kanapala et al., 2019].

Input truncation, unavoidable for long sequences with a low-memory GPU, ignores valuable information, destroying the summary semantic, which is a critical problem also in multi-document summarization [Moro et al., 2022a]. To address this problem, particularly relevant in the legal domain, we propose a new approach for long document summarization: *Semantic Self-Segmentation* (SE3).[1] SE3 creates high-correlated source-target pairs by segmenting long texts into semantically coherent chunks with lengths modulated to fit into the GPU memory, and pairing them with the most similar summary part, enabling transformers to process documents without truncation. This approach works as a data augmentation strategy to cope with the lack of labeled instances, usually addressed with transfer learning methods [Domeniconi et al., 2014b,c, 2015a]. As far as we know, this is the first study on text summarization with limited GPU memories and labeled data scarcity.

Given the complexity of summarizing long legal documents, we experiment on two legal datasets of different domain and content sizes, using SE3 combined with BART [Lewis et al., 2020a] and LED [Beltagy et al., 2020] on a single Titan Xp GPU of 12 GB memory. Results show that SE3 significantly improves the performance of abstractive summarization transformers, even with just a few dozens of labeled training data. Moreover, to analyze where the performance gain comes from, we perform ablation studies and prove the importance of each module of SE3. Finally, we analyze the accuracy of the predicted summaries.

Our contributions are: i) the SE3 method to address long document summarization

[1] Solution website: https://disi-unibo-nlp.github.io/projects/se3

in low-resource regimes; ii) the research advancement on abstractive summarization in legal domains, whose documents are more challenging to analyze, achieving new state-of-the-art results on two datasets.

3.1.2 Related Work

Long Document Summarization Although most solutions focus on short inputs because of the quadratic complexity of transformers, several works presented new approaches to summarize long texts. Celikyilmaz et al. [2018] introduced a hierarchical model that handles the encoding phase through collaborating agents responsible for processing each text subsection. Liu and Chen [2019] and Xu et al. [2020] proposed to exploit the discourse segmentation to extract the salient content for extractive summarization. Gidiotis and Tsoumakas [2020] introduced a divide-and-conquer approach that relies on structured documents to summarize each section independently. Bajaj et al. [2021] compressed long texts by extracting the sentences that best correlate with the summary, adopting an extract-then-abstract paradigm. Grail et al. [2021] and Rohde et al. [2021] modified the standard transformer by adding hierarchical attention layers. Manakul and Gales [2021] showed that applying local self-attention and an explicit content selection improves the performance of large pre-trained quadratic transformers. Cui and Hu [2021] proposed an extractive model that can summarize inputs of arbitrary size without truncation by using a memory network.

Legal Document Summarization Most of the summarization solutions in the legal domain are extractive [Galgani et al., 2015, Tran et al., 2018, Jain et al., 2021a,b, Anand and Wagh, 2022], whereas few studies are abstractive. A first comparative analysis that shows the better performance of abstractive approaches than extractive ones has been proposed by de Vargas Feijó and Moreira [2019], summarizing Brazilian legal rulings. Afterward, Zhang et al. [2020a] achieved new state-of-the-art results on the legal dataset BillSum [Kornilova and Eidelman, 2019] with Pegasus, a transformer-based model with a self-supervised pre-training objective tailored for the abstractive summarization task. In contrast, Huang et al. [2020] extended a pointer-generator network with legal domain-specific knowledge to generate abstractive summaries in the legal public opinion domain.

Low-Resource Summarization About low-resource studies, prior works have only focused on data scarcity. Parida and Motlícek [2019] and Magooda and Litman [2020] proved that augmenting training instances with synthetic data improves the summarization accuracy in low-resource conditions. Bajaj et al. [2021] applied long document summarization with few labeled data, proposing a new method to extract salient sentences from the source. Yu et al. [2021] introduced a new low-resource setting dataset to investigate several adaptive pre-training strategies to cope with the absence of data. Chen and Shuai [2021] proposed meta-transfer learning combined with multiple corpora to improve the accuracy after training models with few labeled data.

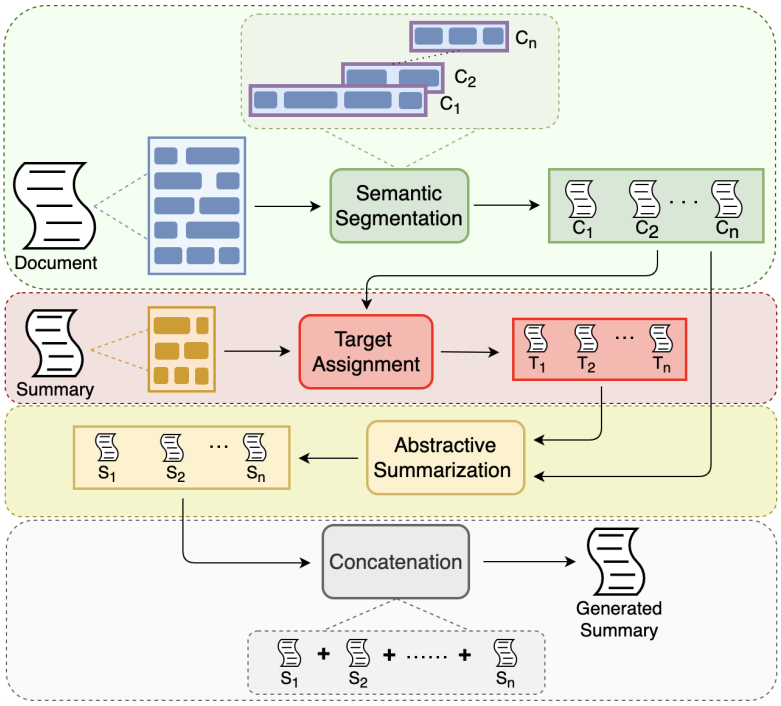


FIGURE 3.1. The overview of SE3 for the abstractive summarization of a long input. First, a document composed of many sentences, i.e., blue rectangles, is segmented into content-wise chunks (green phase). Afterward, each summary sentence, i.e., orange rectangles, is assigned to the most similar chunk, creating new high-correlated source-target pairs (red phase) used to train summarization models (yellow phase). At inference time, the final summary is obtained from concatenating the chunk summaries (gray phase).

Our Work Unlike previous works, we propose a new approach for abstractive long document summarization to address both issues of low-resource regimes, i.e., limited GPU memories and labeled data scarcity, by semantically segmenting long inputs into GPU memory-adaptable chunks.

3.1.3 Preliminary

We provide a better definition of *low-resource regimes*.

Hardware memory scarcity: small and medium-sized organizations can generally afford low-budget GPUs (e.g., 12 GB of memory). These memories limit the training of medium neural models, mostly with long documents, causing "out of memory" exceptions.

Data scarcity: in real scenarios, datasets might consist of only a few dozens of labeled instances. This data shortage limits the learning chances and produces underfit models.

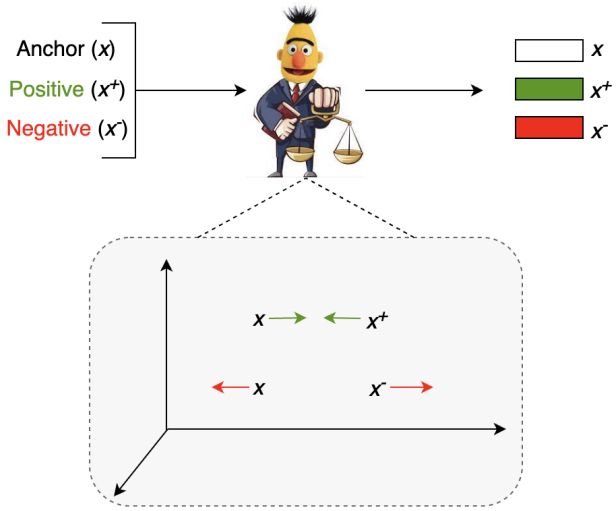


FIGURE 3.2. The metric learning of LEGAL-BERT with the triplet loss. The aim is to create meaningful sentence embeddings by projecting topic-related sentences closer in the vector space and the different ones farther.

3.1.4 Method

Our semantic self-segmentation (SE3) approach for abstractive long document summarization allows fine-tuning transformers on entire long inputs without truncation with limited GPUs (Figure 3.1). Concretely, SE3 segments a long document D into N small chunks, *resolving the hardware memory scarcity issue* since we avoid processing a long text in a single input. Afterward, each sentence of the summary of D is assigned to the most similar input chunk, *resolving the data scarcity issue* since we obtain N chunk-target pairs that augment the training instances.

Two observations motivate this solution: i) truncating input to a fixed length may discard valuable information; ii) in a low-resource scenario, there may also be a lack of labeled data to fine-tune effectively pre-trained models.

3.1.4.1 Semantic Self-Segmentation

We need three elements to train transformers to summarize long inputs without truncation with a limited GPU memory.

The *chunk size* is needed to standardize the chunk content within a range since pre-trained transformers have been trained on fixed sizes, so they struggle to process chunks of very different sizes. Moreover, this range lets users change input size to adapt chunks to the GPU memory available and best leverage the capability of transformers.

A *pre-trained language model* (PLM) is needed to represent the sentences semantically. Thus, as we test SE3 on legal documents, we use LEGAL-BERT [Chalkidis et al., 2020], a BERT model pre-trained on legal corpora.[2] Further, we fine-tune LEGAL-BERT

[2]We could use domain-specific PLMs for other domains, e.g., SCIBERT [Beltagy et al., 2019] for

with metric learning to learn whether two sentences belong to the same section. This learning trains the model to enrich the sentence representation with the thematic meaning, essential for our text segmentation to split sentences. For metric learning, we use a public dataset created in a self-supervised manner [Ein-Dor et al., 2018], as done with paper citations in Moro and Valgimigli [2021], to train models to project sentences of the same section closer in the vector space and the different ones farther (Figure 3.2). We consider two ranking losses in our experiments, i.e., the triplet and the contrastive loss. The triplet loss takes as input a triplet composed of a sentence from a section (anchor, x), a sentence from the same section (positive, x^+), and a sentence from a different section (negative, x^-). The function minimizes the distance between x and x^+ and maximizes the distance between x and x^-, considering a margin m:

$$\mathcal{L}_{\text{triplet}} = \max(||x - x^+|| - ||x - x^-|| + m, 0) \tag{3.1}$$

The contrastive loss takes as input a triplet composed of a sentence from a section x, a second sentence y, and a label l, meaning whether the two sentences belong to the same section (1 if true, 0 otherwise). The loss is as follows:

$$\mathcal{L}_{\text{contrastive}} = l \cdot ||x - y|| + (1 - l) \cdot \max(m - ||x - y||, 0) \tag{3.2}$$

Therefore, our text segmentation algorithm uses the trained language model to produce semantically meaningful sentence embeddings to create the chunks.

A *chunk target* is needed to train abstractive summarization models since we are in a supervised machine learning scenario. For this reason, we assign the most similar part of the summary to the chunks, creating high-correlated source-target pairs. In detail, we apply a syntactic assignment where we pair each sentence of the target summary to the chunk that maximizes the ROUGE-1 precision [Lin, 2004]. Unlike recall, f-measure, or ROUGE-L, we choose such a metric to guarantee a more proper matching for abstractive summaries. The motivations are the following: i) ROUGE-1 checks for uni-gram matching between the summary sentences and the source document, searching the chunk where a summary sentence can be better summarized; ii) the precision metric scores how much content of a summary sentence is within a chunk, searching for the best content coverage.

3.1.4.2 Algorithm

Let $s_{d1}, s_{d2}, ..., s_{dn}$ be the sentences of a document D obtained using the state-of-the-art tokenizer PySBD [Sadvilkar and Neumann, 2020]. Let $s_{s1}, s_{s2}, ..., s_{sm}$ be the sentences of the actual summary of D. Let L_s, U_s be the chunk's lower and upper size, respectively. To create the chunk c_i, along with its target t_i, SE3 performs the following steps (Algorithm 1):

1. Given s_{dj}, if the size of c_i is less than L_s, then add s_{dj} to c_i. This first step does not consider the semantic representation of sentences. However, it is necessary

scientific texts.

Algorithm 1 Semantic Self-Segmentation

Input: *model*; *doc_sentences*; *summary_sentences*
Parameters: $L_s \leftarrow$ lower size; $U_s \leftarrow$ upper size
Output: The chunk-target pairs

Let $chunks = []$
Let $current_chunk = []$
for s_d in *doc_sentences* **do**
  if $len(current_chunk) + len(s_d) < L_s$ **then**
    $current_chunk.append(s_d)$
  else if $len(current_chunk) + len(s_d) > U_s$ **then**
    $chunks.append(current_chunk)$
    $current_chunk \leftarrow []$
  else
    Perform the Semantic Similarity (Algorithm 2)
  end if
end for
$targets \leftarrow$ Perform the Target Assignment (Algorithm 3)
return $(chunks, targets)$

to standardize each chunk to a minimum size to best leverage the capability of transformers since they have been trained on fixed size sequences.

2. Given s_{dj}, if the size of c_i is greater than L_s, and the addition of s_{dj} to c_i does not exceed U_s, we compute the semantic similarity between sentences (Algorithm 2). Otherwise, we create a new chunk c_{i+1} and add s_{dj} to it.

3. To compute the similarity, SE3 first creates the sentence embeddings using the fine-tuned LEGAL-BERT. Afterward, the semantic similarity is calculated between s_{dj} and each sentence within c_i and c_{i+1}. Finally, the similarities are averaged per chunk and compared. In detail, c_{i+1} is created through a look-ahead. More precisely, we perform step 1 until the size of c_{i+1} is at least L_s. Thanks to such a look-ahead, the algorithm does not rely on any hyperparameter similarity threshold. For example, a sentence could be put into the chunk c_i if its semantic similarity with respect to c_i is greater than a fixed value. Instead, we compare the similarity score of the previous chunk with respect to the next one, obtaining an algorithm free from further hyperparameters.

4. Once the chunks have been created, we perform the target assignment (Algorithm 3). Concretely, given s_{sk}, we compare it with each chunk and assign it to the chunk that maximizes the ROUGE-1 precision metric. We then discard chunks without targets at training time.

3.1.4.3 Abstractive Summarization

For experimental purposes, we use both a state-of-the-art quadratic and linear transformer. Their comparison is helpful to analyze how much an efficient transformer can

Algorithm 2 Semantic Similarity

Input: *model*; $s_{dj} \leftarrow$ current sentence; $c_i \leftarrow$ current chunk
Output: Put s_{dj} in the best chunk

Let $c_i \leftarrow [s_{dj-x}, ..., s_{dj-1}]$
Let $c_{i+1} \leftarrow [s_{dj+1}, ..., s_{dj+y}]$
$enc_c_i \leftarrow model.encode(c_i)$
$enc_c_{i+1} \leftarrow model.encode(c_i + 1)$
$score_c_i \leftarrow mean(cosine_sim(enc_c_i, s_{dj}))$
$score_c_{i+1} \leftarrow mean(cosine_sim(enc_c_{i+1}, s_{dj}))$
if $score_c_i > score_c_{i+1}$ **then**
    Put s_j into c_i
else
    Put s_j into c_{i+1}
end if

Algorithm 3 Target Assignment

Input: *summary_sentences*; *chunks*
Output: The targets of the chunks

Let $targets = [t_1 = [], ..., t_w = []]$.
for s_s in *summary_sentences* **do**
    Let $scores = []$.
    for c in *chunks* **do**
        $chunk_score \leftarrow rouge_precision(c, s_s)$
        $scores.append(chunk_score)$
    end for
    $idx \leftarrow argmax(scores)$
    $targets[idx].append(s_s)$
end for
return *targets*

be decisive to improve the summarization accuracy with a limited GPU memory. About the linear transformer, we choose Longformer-Encoder-Decoder [Beltagy et al., 2020], namely LED, because it is the only efficient transformer with a base version public checkpoint. LED replaces the quadratic encoder self-attention using local window attention and global attention. Each token attends to itself and its neighbors in local attention, whereas the first token is connected to everything else in global attention, like in the full attention. About the quadratic transformer, we choose BART [Lewis et al., 2020a] for the following reasons: i) there is a public checkpoint of the base version; ii) it is used as a checkpoint to initialize LED parameters because the latter follows the exact architecture of BART in terms of the number of layers and hidden sizes. The difference is that LED can read more tokens thanks to the linear attention mechanism, making it suitable for processing long documents. We choose the base versions for both models because the large ones do not fit into our GPU memory. For this reason, we make comparisons only with base models.

TABLE 3.1. The dataset statistics. All values are mean over the dataset except for the "# docs" row. We used the LED tokenizer for tokens count and NLTK for words and sentences.

	AustLII		BillSum	
Statistic	Document	Summary	Document	Summary
# sentences	222	14	65	6
# words	7362	667	1592	197
# tokens	7983	722	1673	214
# docs	1754		22218	

3.1.5 Experiments

3.1.5.1 Datasets

We used a dataset comprised of sentence triplets from Wikipedia articles [Ein-Dor et al., 2018] for metric learning. The 1.78M triplets are composed of a sentence pivot, one from the same section, and one from a different section.

We used two legal datasets of different countries (i.e., Australia and the United States) for abstractive summarization. *Australian Legal Case Reports*, referenced as AustLII and publicly downloadable from the UCI archive,[3] is a corpus of around 4000 legal cases from the Federal Court of Australia. We created a target for each document by using the catchphrases provided (i.e., the crucial statements of documents). In detail, we extracted every sentence containing the catchphrase, and we concatenated them to create the actual summary. Since not all documents have catchphrases, we collected 1754 documents, split into 1578 (90%) for training and 176 (10%) for testing. BILLSUM [Kornilova and Eidelman, 2019], downloadable from the Hugging Face library and already split into 18,949 (≈85%) documents for training and 3,269 (≈15%) for testing,[4] consists of 22218 US Congressional Bills with human-written references.

The statistics of the datasets show that the AustLII documents are much longer than the BILLSUM ones (Table 3.1).

3.1.5.2 Experimental Settings

In order to thoroughly evaluate the performance of SE3 in low-resource regimes, the experiments were twofold.

Limited GPU Memory Issue We experimented with six chunk size ranges, expressed in the number of tokens, by segmenting input documents based on the following sizes: *64-128*, *128-256*, *256-512*, *512-1024*, *1024-2048*, and *2048-4096*. About BART, we could not experiment with *1024-2048* and *2048-4096* since it was trained on short documents because of the quadratic memory complexity, so it truncates inputs longer than

[3] https://archive.ics.uci.edu/ml/datasets/Legal+Case+Reports
[4] https://huggingface.co/datasets/billsum

TABLE 3.2. The results of BART with different chunk sizes. Best ROUGE scores are underlined for each max size, i.e., *1024*, *512*, *256*, *128*. The highest are bolded.

System (*MaxLen*)	**AustLII** R1 / R2 / RL	**BillSum** R1 / R2 / RL
Baselines		
PEGASUS$_{\text{BASE}}$	-	51.42/29.68/37.78
BART$_{\text{BASE}}$ (*1024*)	33.51/23.92/27.88	54.42/35.81/41.98
BART$_{\text{BASE}}$ (*512*)	26.61/17.67/21.79	49.84/30.67/37.73
BART$_{\text{BASE}}$ (*256*)	23.87/13.98/18.80	45.99/26.36/34.12
BART$_{\text{BASE}}$ (*128*)	22.11/12.36/17.19	42.32/22.78/31.48
Baselines w/ Se3 - triplet		
BART$_{\text{BASE}}$ (*1024*)	**59.04/52.46/53.67**	57.31/37.85/43.78
BART$_{\text{BASE}}$ (*512*)	53.14/46.44/47.38	55.65/35.73/40.99
BART$_{\text{BASE}}$ (*256*)	44.55/36.50/37.05	51.99/32.63/37.11
BART$_{\text{BASE}}$ (*128*)	37.28/31.42/31.83	44.06/28.69/32.00
Baselines w/ Se3 - contrastive		
BART$_{\text{BASE}}$ (*1024*)	57.96/50.92/52.49	**57.66/38.20/44.11**
BART$_{\text{BASE}}$ (*512*)	52.66/45.71/46.66	55.96/35.82/41.27
BART$_{\text{BASE}}$ (*256*)	45.18/36.82/37.52	52.54/33.00/37.61
BART$_{\text{BASE}}$ (*128*)	37.54/31.89/32.27	44.29/28.90/32.27

1024 tokens. Further, to experiment with two versions of our method, we fine-tuned LEGAL-BERT with both losses, i.e., the triplet and the contrastive loss. To assess whether SE3 allows existing models to achieve a performance gain in low-resource regimes, we used BART and LED as baselines as they were designed, i.e., truncating the input according to each chunk max size without text segmentation. Thus, input sizes and memory requirements are the same, but the solutions with SE3 read the complete document details without truncation.

Labeled Data Scarcity Issue We fine-tuned both models combined with SE3 with 10 and 100 labeled training instances. We experimented only on the BILLSUM dataset to compare our results with recent works on the same low-resource summarization task.

3.1.5.3 Training Details

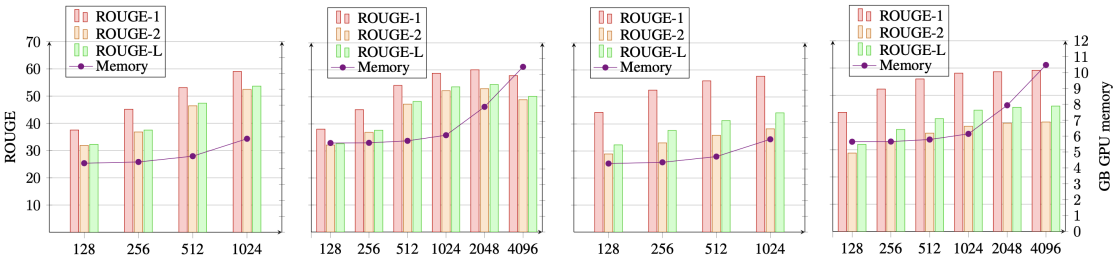
We trained LEGAL-BERT for 1 epoch for metric learning using a batch size of 8 and a learning rate set to 2×10^{-5}. About abstractive summarization, we trained BART and LED for all experiments using the Hugging Face library. All models are fine-tuned for 5 epochs using a batch size of 1 and a learning rate with a linear schedule set to 5×10^{-5}. At inference time, we used 2 as beam size and length penalty.

TABLE 3.3. The results of LED with several chunk sizes. Best ROUGE scores are underlined for each size, i.e., *4096*, *2048*, *1024*, *512*, *256*, *128*. The highest are bolded.

System (*MaxLen*)	**AustLII** R1 / R2 / RL	**BillSum** R1 / R2 / RL
Baselines		
PEGASUS$_{\text{BASE}}$	-	51.42/29.68/37.78
LED$_{\text{BASE}}$ (*4096*)	50.27/39.85/42.04	58.83/39.83/45.71
LED$_{\text{BASE}}$ (*2048*)	42.76/32.20/35.71	58.38/39.37/45.09
LED$_{\text{BASE}}$ (*1024*)	35.20/24.62/28.38	55.32/36.48/42.67
LED$_{\text{BASE}}$ (*512*)	30.47/18.90/23.56	49.96/30.76/37.68
LED$_{\text{BASE}}$ (*256*)	26.77/15.37/20.39	46.76/26.54/34.44
LED$_{\text{BASE}}$ (*128*)	23.78/12.58/18.12	42.75/22.97/31.70
Baselines w/ Se3 - triplet		
LED$_{\text{BASE}}$ (*4096*)	57.89/48.96/50.28	58.51/39.71/45.66
LED$_{\text{BASE}}$ (*2048*)	**60.03/53.03/54.57**	58.38/39.53/45.48
LED$_{\text{BASE}}$ (*1024*)	58.48/52.17/53.48	57.88/38.38/44.15
LED$_{\text{BASE}}$ (*512*)	54.25/47.33/48.32	55.61/35.87/41.04
LED$_{\text{BASE}}$ (*256*)	45.27/36.88/37.68	51.79/32.74/37.09
LED$_{\text{BASE}}$ (*128*)	37.36/31.60/32.09	43.72/28.72/31.88
Baselines w/ Se3 - contrastive		
LED$_{\text{BASE}}$ (*4096*)	57.82/49.06/50.50	**59.18/40.18/46.04**
LED$_{\text{BASE}}$ (*2048*)	60.20/52.40/53.79	58.63/39.77/45.60
LED$_{\text{BASE}}$ (*1024*)	58.75/52.28/53.71	58.11/38.61/44.52
LED$_{\text{BASE}}$ (*512*)	52.37/45.63/46.54	55.99/36.09/41.40
LED$_{\text{BASE}}$ (*256*)	45.35/36.80/37.51	52.28/33.00/37.44
LED$_{\text{BASE}}$ (*128*)	38.07/32.20/32.67	43.74/28.78/31.95

3.1.5.4 Results with Input Longer Than the GPU Memory

Table 3.2 and Table 3.3 summarize BART and LED evaluation results with different chunk sizes on both datasets.



(a) AustLII SE3-BART. (b) AustLII SE3-LED. (c) BILLSUM SE3-BART. (d) BILLSUM SE3-LED.

FIGURE 3.3. The trade-off between performance and memory requirements of SE3 for each max chunk size on both datasets.

TABLE 3.4. Labeled data scarcity summarization on BILLSUM with 10 and 100 training instances. Best values are bolded.

System (*MaxLen*)	**BillSum (10)** R1 / R2 / RL	**BillSum (100)** R1 / R2 / RL
Baselines		
MTL-ABS	41.22/18.61/26.33	45.29/22.74/29.56
$\text{PEGASUS}_{\text{LARGE}}$	40.48/18.49/27.27	44.78/26.40/**34.40**
$\text{BART}_{\text{BASE}}$	39.58/18.94/26.63	44.66/24.87/31.09
LED_{BASE}	41.10/21.15/27.93	47.68/26.98/32.43
Solutions w/ Se3		
$\text{BART}_{\text{BASE}}$ (*1024*)	44.37/21.17/27.57	47.85/26.67/33.36
$\text{BART}_{\text{BASE}}$ (*512*)	46.58/22.03/28.23	49.88/26.84/33.33
$\text{BART}_{\text{BASE}}$ (*256*)	46.50/23.24/28.54	48.17/26.55/31.51
$\text{BART}_{\text{BASE}}$ (*128*)	41.48/22.73/26.37	42.42/25.42/28.98
LED_{BASE} (*4096*)	38.48/19.26/26.36	48.11/26.44/31.91
LED_{BASE} (*2048*)	42.35/20.70/27.12	47.71/26.33/32.12
LED_{BASE} (*1024*)	45.32/22.67/29.12	48.28/26.97/33.46
LED_{BASE} (*512*)	**46.94**/23.04/**29.29**	**50.45**/**27.73**/33.74
LED_{BASE} (*256*)	46.22/**24.32**/29.16	48.13/27.16/31.89
LED_{BASE} (*128*)	40.14/22.76/26.05	40.93/25.29/28.55

Model Performance Comparisons Solutions with SE3 significantly perform the best. In particular, our solution is more effective for the AustLII documents because they are very long, leading to a consistent boost in performance. Indeed, the baselines truncate the input, discarding valuable information in the final summary. Comparison of models shows no performance differences for short inputs. Instead, LED can process longer sequences thanks to the linear complexity of its encoder self-attention, obtaining better results than BART, which can read inputs up to *1024* tokens in size.

Ranking Loss Comparisons The contrastive loss is better than the triplet loss when used for the BILLSUM dataset, differently from the AustLII documents. These results prove that performance mainly depends on the legal content.

Chunk Memory Requirement Comparisons The bigger the chunks, the higher the scores. This result is motivated by the better capability of transformers to process longer sequences. Further, to visualize the scalability of SE3, Figure 3.3 shows the trade-off between the GPU memory used and the model accuracy. The results point out that the best trade-off for both models is *1024* as the max chunk size. LED is trained with a local attention window of *1024* tokens, so it pads inputs if shorter. Therefore, the memory requirements no longer decrease proportionally below this threshold.

TABLE 3.5. The ablations to study how each module of our method contributes to the performance gain. We gradually removed each component of our solution to show the performance drop. Best values are bolded.

	AustLII		BillSum	
Approach	R1 / R2 / RL	R1-P (Train/Test)	R1 / R2 / RL	R1-P (Train/Test)
SE3 (Full)	**58.75/52.28/53.71**	**98.39/98.12**	**58.11/38.61/44.52**	**88.61/88.88**
w/o metric learning	57.11/50.18/51.26	92.87/92.45	57.94/38.44/44.36	88.27/88.44
w/o legal language modeling	56.66/48.95/50.20	98.39/97.97	57.85/38.32/44.03	88.47/88.68
w/o sentence semantics	55.38/47.66/49.05	93.15/93.02	56.65/37.27/43.16	85.62/85.96

TABLE 3.6. The evaluation of the predicted summaries with BERTScore, uni-gram, bi-gram, and trigram uniqueness, and NID. We also provide the values of the reference documents. Best scores are bolded.

	AustLII					BillSum				
System (*MaxLen*)	BERTScore	Uni%	Bi%	Tri%	NID	BERTScore	Uni%	Bi%	Tri%	NID
Reference										
Source	-	22.50	62.21	82.90	28.09	-	25.83	57.82	73.98	30.04
Target	-	51.20	82.37	91.53	23.78	-	57.37	88.33	94.97	21.20
Baselines										
LED$_{BASE}$ (*4096*)	88.59	58.58	91.21	99.68	21.54	90.26	58.76	92.84	99.88	20.77
LED$_{BASE}$ (*2048*)	87.53	60.96	92.23	99.77	21.71	90.20	59.21	92.94	99.87	**20.76**
LED$_{BASE}$ (*1024*)	86.29	60.92	91.75	99.69	22.49	89.82	61.01	93.46	99.88	20.92
LED$_{BASE}$ (*512*)	84.92	59.88	90.41	99.72	23.47	88.93	62.34	93.70	**99.90**	21.38
LED$_{BASE}$ (*256*)	84.26	63.14	92.28	99.79	22.96	88.21	62.36	93.39	99.89	22.08
LED$_{BASE}$ (*128*)	83.46	**67.29**	**94.18**	**99.81**	22.43	87.49	**64.42**	**94.27**	99.88	22.15
Baselines w/ Se3										
LED$_{BASE}$ (*4096*)	89.45	51.59	88.26	97.86	**21.54**	**90.30**	59.00	92.89	99.86	20.77
LED$_{BASE}$ (*2048*)	**89.75**	48.68	86.33	96.78	21.74	90.16	58.87	92.09	98.94	20.84
LED$_{BASE}$ (*1024*)	89.42	44.68	84.00	95.58	21.94	89.79	55.49	89.10	96.35	21.51
LED$_{BASE}$ (*512*)	88.04	41.47	81.20	93.90	22.61	89.04	50.86	85.03	93.47	22.49
LED$_{BASE}$ (*256*)	86.10	39.39	79.62	92.87	23.63	88.11	45.77	80.64	90.54	23.62
LED$_{BASE}$ (*128*)	85.00	33.19	74.16	89.99	24.25	87.12	38.44	74.29	86.86	25.11

3.1.5.5 Results on Labeled Data Scarcity

Table 3.4 shows the performance of labeled data scarcity summarization. We used the first 10 and 100 labeled instances of BILLSUM as done by Zhang et al. [2020a] and Chen and Shuai [2021] with PEGASUS and MTL-ABS, respectively. Our method significantly improves the performance, proving the importance of creating high-correlated source-target pairs in low-resource settings. In detail, the smaller the chunks, the greater the labeled data, allowing transformers to train on more instances. Indeed, we achieved baseline-like results even with models trained on chunk sizes of *64-128* tokens.

3.1.5.6 Ablation Studies

We conducted additional experiments with LED on chunks of *512-1024* tokens, reporting the performance of SE3 after removing 1) metric learning, 2) legal language modeling, and 3) sentence semantic representation (Table 3.5). We notice that excluding either thematic metric learning, legal language modeling, or sentence semantic representation

leads to a performance drop. Training the model without considering the semantic meaning of the sentences in the text segmentation (i.e., ignoring Algorithm 2) leads to the most significant decrease in performance, showing the importance of our semantic self-segmentation algorithm. Indeed, using the sentence representation from BERT improves the accuracy because, without SE3, sentences semantically closer could be split into different chunks, worsening the final summarization. Removing the legal language modeling and the thematic metric learning also decreases the model performance, proving that a domain-specific language model trained on thematic similarity helps create better aligned chunk-target pairs, improving the summarization. In particular, we report the content coverage between chunk-target pairs at training and test time, computed with the average ROUGE-1 precision (R1-P). Results show better alignments using SE3, confirming our method contribution for creating new small high-correlated instances essential for training abstractive summarization transformers in low-resource regimes.

3.1.5.7 Summaries Accuracy

To evaluate the accuracy of the predicted summaries to not rely only on ROUGE, we first used BERTScore [Zhang et al., 2020b] for semantic assessment. Second, we investigated the redundancy due to the independent chunk processing and the final concatenation. To this end, we used the same approaches in Xiao and Carenini [2020]. We first used a *Unique n-gram ratio* to measure n-grams uniqueness. The lower the score, the more redundant the document.

$$Uniq_ngram_ratio = \frac{count(uniq_n_gram)}{count(n_gram)} \tag{3.3}$$

Second, we used the *Normalized Inverse of Diversity (NID)* to capture redundancy by normalizing the unigrams entropy in the document with the maximum possible entropy. The higher the score, the more redundant the document.

$$NID = 1 - \frac{entropy(D)}{log(|D|)} \tag{3.4}$$

Table 3.6 shows the results using LED. BERTScore reports higher results of SE3 on AustLII and similar scores on BILLSUM. Differently, we notice a decrease of n-gram uniqueness with our solution, which is a symbol of more redundancy. Instead, NID scores do not capture such differences.

3.1.6 Conclusion

In this work, we introduced SE3 to address the abstractive long document summarization under low-resource regimes, namely with low-memory GPUs and labeled data scarcity, where the accuracy of existing approaches drops. Thanks to SE3, summarization transformers can process all document details without truncation, achieving also new state-of-the-art results in low-resource scenarios with base models. Moreover, we

proved that the method generates semantically accurate summaries in legal datasets, hence it can be successfully applied to other less complex domains.

We envisage further directions to deal with inputs longer than the GPU memory allows: i) training models to self-annotate cross-chunk salient information through memory-based neural layers [Moro et al., 2018, Cui and Hu, 2021]; ii) extracting relevant texts with term weighting techniques [Domeniconi et al., 2015b] and inter-chunk semantic relations with unsupervised methods [Domeniconi et al., 2014a, 2016b,c] to model interpretable representations with knowledge graph [Frisoni and Moro, 2020, Frisoni et al., 2020b] or relation and event extraction [Domeniconi et al., 2016a, Frisoni et al., 2021].

3.1.7 Broader Impact and Ethical Statement

Summarizing long documents can benefit from our solution, even in small organizations with minimal resources. However, because of the high societal impact of legislation and biases in PLMs [Dey et al., 2020, Liang et al., 2021], domain experts should guide the usage of our method to validate the quality of the inferred summaries. Finally, if the method will be applied to sensitive data, users should also deal with privacy-preserving policies [da Silva et al., 2006].

3.2
Align-Then-Abstract Representation Learning for Low-Resource Summarization

Originally published as: *Gianluca Moro and Luca Ragazzi. Align-then-abstract representation learning for low-resource summarization. Neurocomputing, 548:126356, 2023. doi: 10.1016/J.NEUCOM.2023.126356. URL https://doi.org/10.1016/j.neucom.2023.126356*

Abstract

Generative transformer-based models have achieved state-of-the-art performance in text summarization. Nevertheless, they still struggle in real-world scenarios with long documents when trained in low-resource settings of a few dozen labeled training instances, namely in low-resource summarization (LRS). This paper bridges the gap by addressing two key research challenges when summarizing long documents, i.e., long-input processing and document representation, in one coherent model trained for LRS. Specifically, our novel align-then-abstract representation learning model (ATHENA) jointly trains a segmenter and a summarizer by maximizing the alignment between the chunk-target pairs in output from the text segmentation. Extensive experiments reveal that ATHENA outperforms the current state-of-the-art approaches in LRS on multiple long document summarization datasets from different domains.

3.2.1 Introduction

Abstractive summarization is a natural language processing task of generating a short version of a document while preserving the salient details [Syed et al., 2021]. The training of modern transformer-based solutions [Lewis et al., 2020a, Qi et al., 2020, Raffel et al., 2020, Zhang et al., 2020a] typically requires massive labeled data, which levies obstacles in realistic scenarios of *low-resource summarization* (LRS) distinguished by just a few dozen labeled training instances [Plank, 2016, Hedderich et al., 2021, Moro et al., 2023a]. Of a particular challenge, LRS in the real world regards long documents [Bajaj et al., 2021, Wu et al., 2021a] because producing the ground-truth summary of lengthy texts is expensive, time-consuming, and may demand domain-knowledge proficients. *Long-input processing* is a crux challenge in summarizing extended articles and assumes finding a strategy to address the quadratic memory complexity in the input size of transformer-based models [Vaswani et al., 2017]. Existing methods generally rely on input truncation [Beltagy et al., 2020, Zaheer et al., 2020] or salient content selection [Bajaj et al., 2021]. Nevertheless, the reliance on the input given to the model can undesirably prevent proper *document representation learning*, the quality of which may, in turn, affects the summarization accuracy, especially in low-resource conditions. To date, few efforts have been made to resolve this issue.

This work bridges the gap by reconciling the inherent tension between the two highly dependent problems, i.e., long-input processing and representation learning, in one coherent and synergistic model trained for LRS. Our model, *align-then-abstract representation learning* (ATHENA),[5] jointly trains a text segmentation module with an abstractive summarizer with a novel alignment loss. Notably, we (i) address the computational issues of summarizing lengthy documents by segmenting them into small content-wise chunks, which are synthesized with *fewer memory requirements*, (ii) give summarization models entire inputs *without truncating any information* a priori or selecting a subset of sentences, and (iii) *cope with the scarcity of labeled instances* by implicitly augmenting the training data through text segmentation.

To reckon the effectiveness and generality of our solution, we accomplish comprehensive experiments on multiple public long document summarization datasets from different domains. ATHENA establishes new *state-of-the-art* performance in LRS on all corpora, outperforming previous works significantly.

Our main contributions are as follows:

- To the best of our knowledge, our paper pioneers the exploration of document representation learning for low-resource summarization by presenting a novel *align-then-abstract learning model* (ATHENA) in which the text segmentation is jointly learned to yield better summaries.
- Our proposed approach *nails multiple issues* related to long-input processing: (i) reduces the GPU memory usage, (ii) avoids input truncation, and (iii) works with few labeled training instances.
- ATHENA achieves new *state-of-the-art* results in low-resource summarization on various well-known datasets.

3.2.2 Related Work

Fine-tuning pre-trained models for downstream tasks is a standard strategy, but it is often ineffective if solely dozens of labeled training instances are available. Language models are usually pre-trained with self-supervised learning techniques with numerous unlabeled data [Devlin et al., 2019, Brown et al., 2020, Clark et al., 2020, Lewis et al., 2020a]. Consequently, downstream-specific pre-training strategies [Xiao et al., 2022] have been introduced to create low-resource-oriented models, e.g., PEGASUS [Zhang et al., 2020a]. Nonetheless, with few training samples, pre-trained language models still struggle to adapt to new data from diverse domains [Gururangan et al., 2020]. For this reason, several approaches have been proposed to tackle the limited availability of labeled instances. Prompt-based methods [Li and Liang, 2021, Liu et al., 2022b] tune continuous prompts to adapt quickly to new tasks with few examples. Other works [Parida and Motlícek, 2019, Magooda and Litman, 2020] applied synthetic data augmentation, enhancing the summarization accuracy in low-resource conditions but only

[5] https://disi-unibo-nlp.github.io/projects/athena/

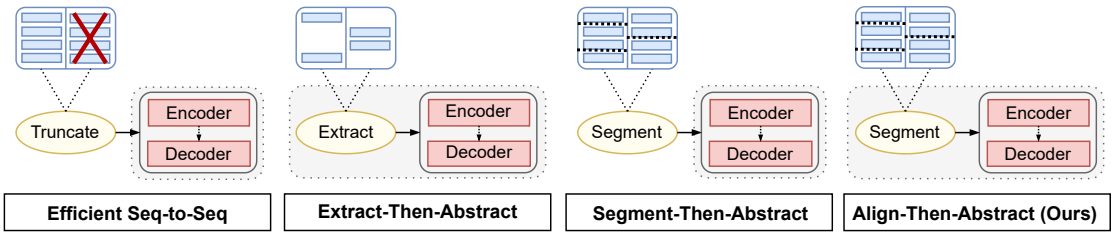


FIGURE 3.4. The establish model architectures for long document summarization. Gray parts indicate the learning modules. The "align-then-abstract" is our proposed architecture.

experimenting on short texts of max 400 and 200 tokens, respectively. Conversely, to mimic the real-world LRS scenario over long documents, Bajaj et al. [2021] proposed an extract-then-abstract approach to provide exclusively salient sentences to a pre-trained model. Despite its effectiveness, this solution involves a two-stage training, in which the summarization model does not process all information in the long input. Finally, Chen and Shuai [2021] introduced a meta-transfer learning technique that augments the training data with multiple similar corpora.

Unlike prior contributions, we experiment with multiple public long document summarization datasets from different domains using a base model without synthetic data augmentation.

3.2.3 Background

Problem Definition We define the problem of *long document summarization* with the following setup. The input is a lengthy text $\mathcal{X} = \{x_1, \ldots, x_x\}$ coupled with its corresponding summary $\mathcal{Y} = \{y_1, \ldots, y_y\}$, where each $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ is a token. The standard training algorithm adopts the cross-entropy loss, which requires the model to predict the next token y_i of the target summary given $\mathcal{X}$ and the previous target tokens $y_{1:i-1}$, as follows:

$$\mathcal{L}_{ce} = -\sum_{i=1}^{|\mathcal{Y}|} \log p_\theta(y_i | y_{1:i-1}, \mathcal{X}) \tag{3.5}$$

where θ indicates the model parameters and p is the predicted probability over the vocabulary.

Model Architectures The number of input tokens could be potentially significant (e.g., >10,000). Therefore, $\mathcal{X}$ cannot be processed at its full size with current quadratic transformer-based models and commodity hardware environments. For this reason, several model architectures have been proposed to handle long-input processing: (i) *Efficient sequence-to-sequence* reads more input tokens thanks to sparse attention mechanisms with linear complexity in the input size [Beltagy et al., 2020, Zaheer et al., 2020, Huang et al., 2021, Guo et al., 2022a, Phang et al., 2023]. (ii) *Extract-then-abstract* lessens the input size by supplying just a subset of the source to the summarization model [Bajaj

et al., 2021, Mao et al., 2022, Moro et al., 2022a]. (iii) *Segment-then-abstract* divides the input into sections, independently summarized and concatenated to produce the final summary [Gidiotis and Tsoumakas, 2020, Wu et al., 2021a, Moro and Ragazzi, 2022, Zhang et al., 2022d, Moro et al., 2023d].

3.2.4 Method

In this section, we describe our *align-then-abstract representation learning model* (ATHENA) in detail. In a nutshell, ATHENA comprises two key collaborating modules tackling long-input processing (Section 3.2.4.1) and document representation learning (Section 3.2.4.2). The two components work jointly to learn the best text segmentation that yields better summaries. Our model differs from existing ones thanks to an end-to-end learning solution in which segmentation and summarization cooperate to generate the synthesis. An overview of our proposed and existing architectures is shown in Figure 3.4.

3.2.4.1 Long-Input Processing

In long document summarization, the source length, in terms of the number of tokens, may exceed the limit summarization models can consume (e.g., BART [Lewis et al., 2020a] truncates inputs longer than 1024 tokens). Nevertheless, we argue that the whole document information can contribute to the final summary. To this aim, our ATHENA model is trained to synthesize a long input by segmenting it into small coherent chunks, learning end-to-end the best document segmentation, thus summarizing the chunks and concatenating the chunk-level summaries to produce the final prediction. Consequently, our model can read the entire document without truncating any information or relying on a subset of snippets.

Source Segmentation To segment a long input into small chunks, we leverage the SE3 algorithm [Moro and Ragazzi, 2022]. This unsupervised method uses a BERT-based model to semantically represent the sentences and create the chunks based on their meaning.[6] Unlike SE3, which employs a frozen BERT, we follow the same algorithm but introduce a novel loss (Section 3.2.4.2) to train the segmenter end-to-end to uncover the best document segmentation that improves the summarization accuracy. Overall, ATHENA segments a long input $\mathcal{X}$ into n not-overlapping chunks $\{\bar{\mathcal{X}}_1, \ldots, \bar{\mathcal{X}}_n\}$, each with a number of tokens $\leq \mathcal{M}$, corresponding to the max input size summarization models can process (e.g., $\mathcal{M} = 1024$ for BART).

Target Alignment We align each sentence $y_j \in \mathcal{Y}$ to the chunk that can better summarize it, yielding new high-correlated instances $(\bar{\mathcal{X}}, \bar{\mathcal{Y}})$. More precisely, each sentence y_j is assigned to the chunk $\bar{\mathcal{X}}_i$ that maximizes the formula:

$$\mathcal{R}_p^1(\bar{\mathcal{X}}_i, \bar{\mathcal{Y}}_i \circ y_j) \tag{3.6}$$

[6] Please refer to the original paper [Moro and Ragazzi, 2022] for details.

where $\mathcal{R}_p^1$ stands for the ROUGE-1 precision metric [Lin, 2004] and $\circ$ indicates the concatenation of previous target sentences already assigned to $\bar{\mathcal{X}}_i$. Note that this matching algorithm does not assure $\bar{\mathcal{Y}}_i \neq \emptyset$, noticeable if the number of chunks exceeds the summary sentences. Yet, we aim to create high-correlated chunk-target pairs $(\bar{\mathcal{X}}, \bar{\mathcal{Y}})$ for the summarization model (see Section 3.2.5.8).

3.2.4.2 Document Representation Learning

Our method is trained end-to-end to maximize the conditional probability of generating $\bar{\mathcal{Y}}_i$ from $\bar{\mathcal{X}}_i$, where $(\bar{\mathcal{X}}, \bar{\mathcal{Y}})$ is the set of chunk-target pairs produced by the segmentation and alignment modules. To jointly train the segmenter, we propose *alignment loss* to teach the model to segment better $(\mathcal{X}, \mathcal{Y})$ into more *aligned* pairs $(\bar{\mathcal{X}}, \bar{\mathcal{Y}})$, learning the best document representation.

Alignment Loss The segmenter is trained to maximize the alignment between each chunk-target pair $(\bar{\mathcal{X}}_i, \bar{\mathcal{Y}}_i)$ in terms of semantic content coverage, encouraging the model to locate the best text segmentation that improves the summarization. In detail, the alignment loss $\mathcal{L}^{\eta}_{align}$ computes the cosine similarity between $(\bar{\mathcal{X}}^e, \bar{\mathcal{Y}}^e)$, where e denotes the embedding representation of each $(\bar{\mathcal{X}}_i, \bar{\mathcal{Y}}_i) \in (\bar{\mathcal{X}}, \bar{\mathcal{Y}})$. Specifically, to obtain $\bar{\mathcal{X}}^e$, we generate the sentence embeddings with the segmenter (η denotes its parameters) by computing the mean pooling operation over the token embeddings of each sentence [Reimers and Gurevych, 2019]. Afterward, we calculate the mean over the output vectors to obtain a single embedding for each chunk and target. With this loss, the model learns to maximize the alignment between the chunk-target pairs, thus learning to better segment and represent the documents. More precisely, the weights of the model updated during this learning are the same used for segmenting the document sentences into chunks. Hence, we train the model to segment the document better by maximizing the alignment between the chunk-target pairs created after the segmentation. The alignment loss is the following:

$$\mathcal{L}^{\eta}_{align} = 1 - \frac{\bar{\mathcal{X}}^e \cdot \bar{\mathcal{Y}}^e}{|\bar{\mathcal{X}}^e||\bar{\mathcal{Y}}^e|} \tag{3.7}$$

Summarization Loss The summarization module takes as input the chunk-target pairs and is trained to generate the next output token for each target by minimizing the negative log-likelihood with the following function:

$$\mathcal{L}^{\gamma}_{summ} = -\frac{1}{|\bar{\mathcal{Y}}|}\sum_{t=1}^{|\bar{\mathcal{Y}}|} \log p(y_t | y_{1:t-1}, \bar{\mathcal{X}}) \tag{3.8}$$

where γ are the parameters of the summarization module, $\bar{\mathcal{X}}$ is the input chunk, and $y_{1:t}$ are the tokens from position 1 to t of the target $\bar{\mathcal{Y}}$. Note that, for the training process, we take only the chunk-target pairs $(\bar{\mathcal{X}}_i, \bar{\mathcal{Y}}_i)$ such that $\bar{\mathcal{Y}}_i \neq \emptyset$. In contrast, we consider all the chunks at inference time.

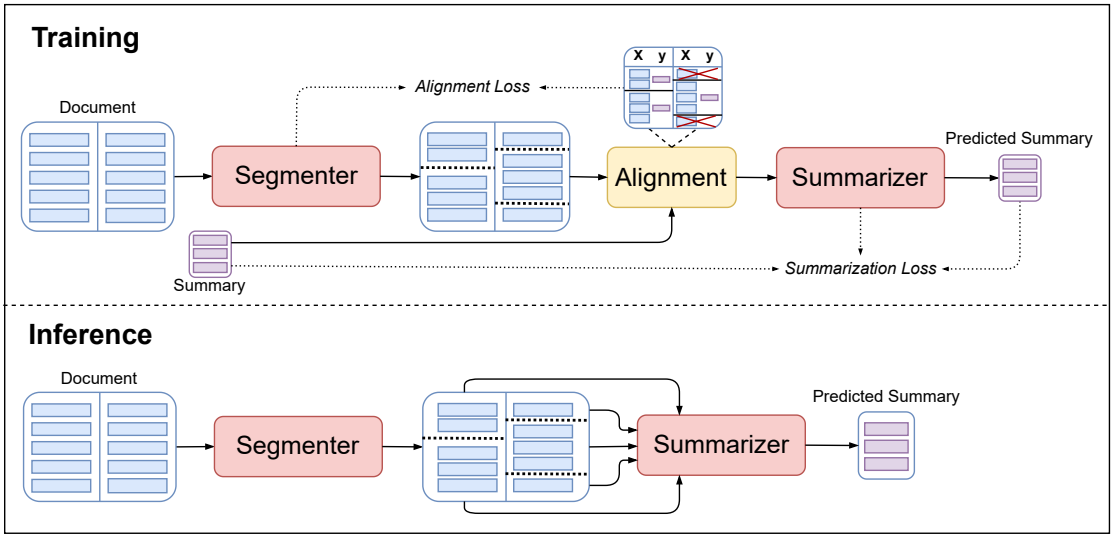


FIGURE 3.5. The illustration of ATHENA at training and inference time. The input is a long document and the output is a short summary.

Algorithm 4 Align-Then-Abstract Learning

Input:
$\mathcal{X} = \{x_1, \ldots, x_x\} \leftarrow$ Input; $\mathcal{Y} = \{y_1, \ldots, y_y\} \leftarrow$ Output; BERT; BART
Training:
1: $\bar{\mathcal{X}} \leftarrow Segmentation(\mathcal{X}, \text{BERT})$ ▷ Chunks
2: $\bar{\mathcal{X}}, \bar{\mathcal{Y}} \leftarrow Alignment(\bar{\mathcal{X}}, \mathcal{Y})$ ▷ Chunk-target pairs
3: $AlignmentLoss(\bar{\mathcal{X}}, \bar{\mathcal{Y}}, \text{BERT})$ ▷ Backpropagation over BERT
4: $SummarizationLoss(\bar{\mathcal{X}}, \bar{\mathcal{Y}}, \text{BART})$ ▷ Backpropagation over BART

3.2.4.3 Training Objective

The overall training objective of our solution is the following:

$$\mathcal{L}^{\eta,\gamma}(\hat{\mathcal{Y}}_i, \mathcal{Y}_i) = \mathcal{L}^{\eta}_{align} + \mathcal{L}^{\gamma}_{summ} \tag{3.9}$$

The whole model is trained end-to-end with an align-then-abstract approach to segment the source into content-wise chunks and summarize each of them. Concretely, the segmenter is optimized with the alignment loss, whereas the summarizer is optimized with the summarization loss (Algorithm 4). Moreover, we involve an update step of the model weights with a dynamic mini-batch gradient descent equal to the number of chunks per instance, formally defined as follows:

$$\theta_j = \theta_j - \epsilon \frac{1}{n} \sum_{i=n \cdot k}^{n(k+1)} \nabla_{\theta_j} \mathcal{L}(\hat{\mathcal{Y}}_i, \mathcal{Y}_i) \tag{3.10}$$

where $\theta = \eta + \gamma$ are the parameters of the full model, $n = |\bar{\mathcal{X}}|_i$ is the number of

TABLE 3.7. The long document summarization datasets used as testbeds. Statistics include corpus size, number of source texts per instance, number of total words in source and target texts, and source-target coverage, density, and compression ratio of words [Grusky et al., 2018]. Except for the number of samples, all reported values are averaged across all instances.

			Source		Target		Source → Target		
Dataset	**Domain**	**Samples**	Words	Sents	Words	Sents	Coverage	Density	Compress
BILLSUM [Kornilova and Eidelman, 2019]	Bills	23,455	1673.25	46.49	212.96	4.99	0.90	6.86	12.21
PUBMED [Cohan et al., 2018]	Biomedical	130,397	3166.12	205.39	98.87	7.36	0.89	5.98	16.69
GOVREPORT [Huang et al., 2021]	Legal	19,463	8765.01	298.69	556.30	18.10	0.94	9.08	17.85

chunks of the i-th document, and $k = \{1, \frac{\mathcal{N}}{n}\}$ is the number of update steps, with $\mathcal{N}$ equal to the amount of training samples. In this way, the gradients are (i) computed for each document chunk, implicitly augmenting the training instances, (ii) averaged per document, and (iii) descended after each instance.

Figure 3.5 illustrates our proposed solution. Technically, while the input is a single training instance, the summarizer reads more labeled samples (3 in the example) produced by the alignment module. At inference time, the final summary is obtained by concatenating the predicted chunk-level summaries. Note that the chunks without an assigned target are not considered during training.

3.2.5 Experiments

3.2.5.1 Setup

To assess the performance of our solution, we simulate a low-resource scenario with labeled data scarcity, adopting the identical experimental setup in previous comparative works on low-resource summarization [Zhang et al., 2020a, Chen and Shuai, 2021] for a fair comparison and reproducibility. Specifically, we train our model with all datasets' first 10 and 100 training instances.

3.2.5.2 Datasets

We contemplate the following well-known public long document summarization datasets from different domains and text sizes as evaluation benchmarks. Key measurements are reported in Table 3.7.[7]

- BILLSUM [Kornilova and Eidelman, 2019] consists of 22K U.S. congressional bills from the 103rd-115th (1993-2018) sessions of Congress.
- PUBMED [Cohan et al., 2018] comprises 133K biomedical publications from PubMed.
- GOVREPORT [Huang et al., 2021] includes 19K U.S. government reports.

[7]All datasets can be accessed through Hugging Face: https://huggingface.co/datasets

3.2.5.3 Baselines

We compare ATHENA with cutting-edge baselines:

BART [Lewis et al., 2020a] is the state-of-the-art denoising sequence-to-sequence pre-trained model for various text generation tasks. Since ATHENA is built upon BART, this comparison can reveal whether the segmentation approach is meaningful in long-input processing. We report the results of BART-base.

PEGASUS [Zhang et al., 2020a] is a transformer-based model with a summarization specific pre-training that helps fast adapting with few labeled data. As there is no public checkpoint of the base version, we include the results of PEGASUS-large.

MTL-ABS [Chen and Shuai, 2021] is a meta-transfer learning approach for LRS that cope with data scarcity by augmenting the training data with multiple similar corpora.

LED [Beltagy et al., 2020] is a state-of-the-art efficient transformer built upon BART with a self-attention mechanism that scales linearly in the input size, allowing the model to process long sequences. We employ LED-base.

LONGT5 [Guo et al., 2022a] is a powerful pre-trained model with sparse attention built upon T5. We use LONGT5-base.

3.2.5.4 Evaluation Metrics

Although human evaluation is deemed the gold standard for estimating model accuracy, it is prohibitively expensive, and recent research has even shown some shortcomings [Schoch et al., 2020]. For these reasons, we embrace automatic evaluation metrics, assessing the inferred summaries from different perspectives.

Lexical Overlap We use ROUGE-{1,2,L} F1 scores against reference summaries, reporting R-1 and R-2 for informativeness and R-L for fluency.[8] Additionally, inspired by Moro et al. [2023b], we also compute $\mathcal{R} = {}^{\text{avg(R-1,R-2,R-L)}}/_{1+\sigma_r^2}$ to derive an aggregated judgment $\in [0, 1]$ (the higher, the better) that penalizes generations with heterogeneous results across the ROUGE dimensions.

Semantic Similarity We report BERTScore F1 (BS) [Zhang et al., 2020b],[9] which computes the contextual similarity between a candidate and its reference summary. We use this metric to assess the model performance in the ablation studies.

3.2.5.5 Implementation Details

Pre-trained Models The summarizer is initialized with BART-base [Lewis et al., 2020a] weights, whereas the segmenter is initialized with a pre-trained sentence embedding model based on BERT-small [Devlin et al., 2019].[10] Technically, the segmenter is

[8] We use ROUGE provided by Hugging Face: `https://huggingface.co/metrics/rouge`.
[9] We use BERTScore from Hugging Face: `https://huggingface.co/metrics/bertscore`.
[10] The checkpoint is public in Hugging Face: `sentence-transformers/all-MiniLM-L6-v2`.

TABLE 3.8. Low-resource summarization performance of ATHENA on all target datasets. † means that the results are from the original papers. The other models are fine-tuned with the same training details of ATHENA. Best scores on each dataset are bolded.

		BillSum				PubMed				GovReport			
Model	**Sample**	**R-1**$_{f1}$	**R-2**$_{f1}$	**R-L**$_{f1}$	$\mathcal{R}$	**R-1**$_{f1}$	**R-2**$_{f1}$	**R-L**$_{f1}$	$\mathcal{R}$	**R-1**$_{f1}$	**R-2**$_{f1}$	**R-L**$_{f1}$	$\mathcal{R}$
PEGASUS†	10	40.48	18.49	27.27	28.51	33.31	10.58	20.05	21.13	-	-	-	-
	100	44.78	26.40	34.40	35.00	34.05	12.75	21.12	22.47	-	-	-	-
MTL-ABS†	10	41.22	18.61	26.33	28.47	34.08	10.05	18.66	20.73	-	-	-	-
	100	45.29	22.74	29.56	32.24	35.19	11.44	19.89	21.96	-	-	-	-
BART	10	45.59	22.83	29.05	32.19	38.07	12.49	19.91	23.22	48.63	18.48	21.23	28.91
	100	49.74	27.15	32.93	36.27	40.29	13.51	21.85	24.90	43.77	13.39	19.91	25.26
LED	10	45.57	22.89	29.05	32.21	38.43	11.90	19.93	23.13	52.73	18.54	20.73	29.94
	100	48.44	26.62	32.21	35.45	42.07	14.34	22.22	25.86	55.35	20.94	22.02	31.95
LONGT5	10	42.75	18.98	25.31	28.72	36.06	10.54	18.31	21.39	50.12	16.55	19.40	28.04
	100	42.79	19.00	25.31	28.74	36.16	10.65	18.36	21.48	50.06	16.51	19.38	28.00
Athena	10	**47.57**	**24.14**	**30.35**	**33.69**	**40.98**	**13.35**	**21.39**	**24.90**	**54.95**	**19.98**	**22.24**	**31.58**
	100	**51.59**	**29.36**	**35.04**	**38.32**	**42.46**	**14.72**	**22.63**	**26.25**	**56.85**	**22.06**	**23.19**	**33.17**

based on a siamese network and has already been fine-tuned using a contrastive learning objective for the semantic textual similarity task.

Training We train on all datasets with mixed precision for 20 epochs, saving the model that perform best on the validation set.[11] We apply gradient checkpointing to save memory, use the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$, and set the learning rate to 3e-5. We consider a chunk size between 256 and 1024 tokens for text segmentation. We eventually set the seed to 42 for reproducibility.

Inference We set the beam width to 5 and use the following summary size (min-max) based on experiments and statistics reported in Table 3.7: BILLSUM (100-300), PUBMED (100-300), GOVREPORT (500-1000). Finally, we utilize an n-grams penalty of 5 for GOVREPORT and 3 for the other datasets.

Hardware Each experiment is run on a single RTX 3090 GPU of 24GB memory with PyTorch [Paszke et al., 2019] in a workstation with 64GB of RAM and an Intel® Core™ i9-10900X1080 CPU @ 3.70GHz.

3.2.5.6 Results and Discussion

The evaluation results are reported in Table 3.8. ATHENA achieves new state-of-the-art ROUGE scores in low-resource conditions with a wide margin on BILLSUM, PUBMED, and GOVREPORT.

Capacity to Summarize Long Inputs The results suggest the effectiveness of the align-then-abstract approach for long document summarization in low-resource condi-

[11] We use only the first 10 validation samples to simulate real-world low-resource conditions.

TABLE 3.9. Chunk-target alignment in ATHENA of a random sample from BILLSUM.

Chunk 1

Section 1. short title. this act may be cited as the ""new idea "" . sec. 2. clarification that wages paid to unauthorized aliens may not be deducted from gross income. in general. subsection (c) of section 162 of the internal revenue code of 1986 is amended by adding at the end the following new paragraph: wages paid to or on behalf of unauthorized aliens . in general. no deduction shall be allowed under subsection (a) for any wage paid to or on behalf of an unauthorized alien , as defined under section 274a(h)(3) of the immigration and nationality act (8 u. s. c. 1324a(h). *[. . .]*

Target 1

New idea - amends the internal revenue code to deny a tax deduction for wages and benefits paid to or on behalf of an unauthorized alien.

Chunk 2

[. . .]

the commissioner of social security, the secretary of the department of homeland security, and the secretary of the treasury, shall jointly establish a program to share information among such agencies that may or could lead to the identification of unauthorized aliens (as defined under section 274a(h), including any no-match letter, any information in the earnings suspense file, and any information in the investigation and enforcement of section 162(c)(4) of the internal revenue code of 1986. disclosure by secretary of the treasury. in general. subsection (i) of section 6103 of the internal revenue code of 1986 is amended by adding at the end the following new paragraph: payment of wages to unauthorized aliens. upon request from the commissioner of the social security administration or the secretary of the department of homeland security, the secretary shall disclose to officers and employees of such administration or department taxpayer identity information of employers who paid wages with respect to which a deduction was not allowed by reason of section 162(c)(4), and taxpayer identity information of individuals to whom such wages were paid, for purposes of carrying out any enforcement activities of such administration or department with respect to such employers or individuals.". record keeping. paragraph (4) of section 6103(p) of such code is amended by striking "(5), or (7)" in the matter preceding subparagraph (a) and inserting , (7), or (9)", and by striking "(5) or (7)" in subparagraph (f) and inserting "(5), (7), or (9)".

Target 2

directs the commissioner of social security and the secretaries of homeland security and the treasury to jointly establish a program to share information that may lead to the identification of unauthorized aliens. requires the secretary of the treasury to provide taxpayer identity information to the commissioner of social security and the secretary of homeland security on employers who paid nondeductible wages to unauthorized aliens and on the aliens to whom such wages were paid.

Chunk 3

[. . .]

section 401 of the illegal immigration reform and immigrant responsibility act of 1996 is amended by striking the last sentence. application to current employees. voluntary election. the first sentence of section 402(a) of such act is amended to read as follows: "any person or other entity that conducts any hiring in a state or employs any individuals in a state may elect to participate in a pilot program.". benefit of rebuttable presumption. paragraph (1) of section 402(b) of such act is amended by adding at the end the following: "if a person or other entity is participating in a pilot program and obtains confirmation of identity and employment eligibility in compliance with the terms and conditions of the program with respect to individuals employed by the person or entity, the person or entity has established a rebuttable presumption that the person or entity has not violated section 274a(a)(2) with respect to such individuals." .

Target 3

amends the illegal immigration reform and immigrant responsibility act of 1996 to: (1) make permanent the pilot program for verifying the employment eligibility of alien workers, (2) apply such program to current employees in addition to new hires, and (3) establish a rebuttable presumption that employers who participate in the pilot program have not violated the prohibition against continued employment of unauthorized aliens.

tions. Indeed, segmenting a long input into small content-wise chunks allows existing models to summarize very lengthy documents by processing small chunks, extending

Table 3.10. ROUGE scores of Athena using different subsets of the training set.

Dataset	Instances	#1	#2	#3	#4	#5
BillSum	10	47.57/24.14/30.35	47.88/24.85/30.81	48.57/25.40/31.07	47.36/23.77/29.56	47.54/24.37/30.29
	100	51.59/29.36/35.04	51.04/28.96/34.87	50.66/28.38/34.42	51.23/28.98/34.69	51.09/28.65/34.76
PubMed	10	40.98/13.35/21.39	41.00/13.16/20.96	41.53/13.37/21.62	41.80/13.63/21.72	40.61/12.86/21.00
	100	42.46/14.72/22.63	43.09/14.95/22.68	43.41/15.52/23.01	43.00/15.23/22.85	43.39/15.40/22.90
GovReport	10	54.95/19.98/22.24	55.72/20.76/22.74	55.58/20.58/22.46	54.96/20.18/22.11	56.02/20.99/22.63
	100	56.85/22.06/23.19	56.94/22.19/23.23	56.39/21.72/23.05	57.06/22.06/23.19	56.95/22.05/23.25

the input size that the model can handle. In this way, the summarization phase reads all document details, handling the long input at its full length, avoiding prior input truncation or processing only a subset of pre-selected sentences.

Capacity to Adapt to Data Scarcity We show the high capability of our model to synthesize long sequences in a low-resource data scenario. During the training phase, the segmenter creates small high-correlated chunk-target pairs and feeds the summarization module many high-quality training samples that augment the number of training instances and the model's focus during at learning time. Table 3.9 illustrates a qualitative example of chunk-target pairs correlation in Athena, namely the training data and their labels.

Capacity to Handle Multiple Domains Our model attains new state-of-the-art results on a comprehensive collection of datasets, indicating a high potential of adapting to different dictionaries and language styles.

Efficiency and Effectiveness Athena is built upon small and base models, so it is memory-efficient and more practical to use in small- and medium-sized organizations that cannot afford high-budget GPU memories (i.e., more than 12GB). Regardless, despite its small number of parameters, Athena achieves state-of-the-art results, proving its effectiveness and a conceivable additional gain with large models and more GB of GPU memory available.

Table 3.11. Analyzes on the segmented chunks at inference time. All values are averaged.

Dataset	Instances	# Chunks	Chunk Size
BillSum	10	4.9	436.0
	100	4.6	451.0
PubMed	10	9.4	372.7
	100	9.1	383.2
GovReport	10	24.5	361.1
	100	24.2	365.2

TABLE 3.12. Running time and GPU memory requirement of models.

Dataset	Complexity	Bart	Led	LongT5	Athena (Ours)
BILLSUM	Time (s)	1.0	2.4	5.0	4.7
	GPU (GB)	5.5	7.5	8.1	5.7
PUBMED	Time (s)	1.2	3.5	6.0	9.0
	GPU (GB)	5.7	11.2	14.1	6.8
GOVREPORT	Time (s)	1.4	7.0	10.2	30.2
	GPU (GB)	6.8	15.9	21.1	7.2

3.2.5.7 Subset Analysis

Unlike prior works, we believe that the selection of samples plays a vital role in the final results in low-resource regimes. For this reason, we conduct further experiments on multiple subsets of the training sets. Technically, we use the first 5 not-overlapping subsets with 10 and 100 instances within each subset to assess if the performance of our proposed model remains stable or highly depends on the input data. Table 3.10 reports the high similarity of the results despite the different training subsets.

3.2.5.8 Segmentation Analysis

We first study how the segmentation affects the number and the size of the chunks at inference time. Table 3.11 shows the mean number of chunks per corpus, which is really high for the GOVREPORT dataset. We notice that the documents are segmented into chunks of about 400 tokens in length for all datasets, despite the distinct text sizes.

3.2.5.9 Complexity Analysis

Our model ATHENA has a quadratic memory growth w.r.t. the chunk size. Therefore, the space complexity to summarize the entire input text is $\mathcal{O}(L_c^2)$, where L_c is the max chunk size (or the model max input length). Table 3.12 reports the running time and the GPU memory requirement of models on all datasets trained on 10 instances for 1 epoch. The more training time required for ATHENA is due to its capability to read all the long document chunk by chunk, allowing a low-memory occupation.

3.2.5.10 Ablation Studies

We conduct ablation studies to investigate the effectiveness of the modules of our solution. In detail, we report the performance of ATHENA after removing alignment loss and generation loss. The results are summarized in Table 3.13.

We notice that excluding the generation loss (w/o $\mathcal{L}_{gen}$) leads to the most significant loss in performance. Nonetheless, we still achieve competitive results, demonstrating the excellent capability of our solution architecture. Training the model without considering the alignment loss (w/o $\mathcal{L}_{align}$) decreases the performance, showing the importance of creating high-correlated samples in low-resource conditions.

TABLE 3.13. Ablation studies to validate the effectiveness of the full method ($\mathcal{L} = \mathcal{L}_{align} + \mathcal{L}_{gen}$). Best results are bolded.

	10					100				
	R-1$_{f1}$	**R-2**$_{f1}$	**R-L**$_{f1}$	$\mathcal{R}$	**BS**$_{f1}$	**R-1**$_{f1}$	**R-2**$_{f1}$	**R-L**$_{f1}$	$\mathcal{R}$	**BS**$_{f1}$
BillSum										
Full	**47.57**	**24.14**	**30.35**	**33.69**	**86.26**	**51.59**	**29.36**	**35.04**	**38.32**	**87.65**
w/o $\mathcal{L}_{align}$	46.98	23.56	29.69	33.08	86.07	51.11	28.92	34.90	37.98	87.59
w/o $\mathcal{L}_{gen}$	42.07	18.95	25.14	28.45	83.97	42.03	18.92	25.16	28.43	83.97
PubMed										
Full	**40.98**	**13.35**	**21.39**	**24.90**	**83.86**	**42.46**	**14.72**	**22.63**	**26.25**	**84.29**
w/o $\mathcal{L}_{align}$	40.63	12.92	21.03	24.53	83.64	42.30	14.67	22.54	26.15	84.21
w/o $\mathcal{L}_{gen}$	39.87	12.34	20.29	23.85	83.35	39.81	12.31	20.28	23.82	83.34
GovReport										
Full	54.95	19.98	**22.24**	31.58	84.81	56.85	**22.06**	**23.19**	**33.17**	**85.13**
w/o $\mathcal{L}_{align}$	**55.78**	**20.36**	22.23	**31.94**	**84.95**	**56.94**	21.93	23.13	33.13	85.12
w/o $\mathcal{L}_{gen}$	54.50	19.93	21.92	31.33	84.65	54.28	19.67	21.84	31.15	84.61

3.2.5.11 Comparison with Se3

The architecture of our solution is related to but differs significantly from SE3 [Moro and Ragazzi, 2022]. The similarity only lies in using the segmentation algorithm to split a long input into multiple chunks. However, unlike our dynamic chunk creation through learning, the segmentation module in SE3 is frozen with pre-selected chunk sizes. Table 3.14 reports the results of SE3 on the BILLSUM dataset to further prove the importance of learning how to better segment a long input.

3.2.6 Conclusion

In this paper, we propose ATHENA, a novel approach for long document summarization in low-resource conditions, namely with just dozens of labeled training instances available, which is a real-world scenario. ATHENA is trained end-to-end on an align-then-abstract representation learning to better segment a long input to (i) create small content-wise

TABLE 3.14. Comparison with SE3 on the BILLSUM dataset (the results are taken from the original paper [Moro and Ragazzi, 2022]). Best values are bolded.

	BillSum (10)				BillSum (100)			
Model	**R-1**$_{f1}$	**R-2**$_{f1}$	**R-L**$_{f1}$	$\mathcal{R}$	**R-1**$_{f1}$	**R-2**$_{f1}$	**R-L**$_{f1}$	$\mathcal{R}$
SE3 (*1024*)	44.37	21.17	27.57	30.74	47.85	26.67	33.36	35.68
SE3 (*512*)	46.58	22.03	28.23	31.93	49.88	26.84	33.33	36.34
SE3 (*256*)	46.50	23.24	28.54	32.44	48.17	26.55	31.51	35.11
SE3 (*128*)	41.48	22.73	26.37	30.00	42.42	25.42	28.98	32.10
ATHENA (*dynamic*)	**47.57**	**24.14**	**30.35**	**33.69**	**51.59**	**29.36**	**35.04**	**38.32**

chunks processable with fewer memory requirements, (ii) read the long text in its full size, and (iii) create high-correlated samples to augment the data with high-quality source-target instances. We demonstrate the effectiveness of our solution by benchmarking three datasets of different domains, significantly outperforming the current state-of-the-art in low-resource summarization on all datasets.

For future works, we suggest investigating the following approaches to better model chunks creation: (i) memory-based operations [Moro et al., 2018] from unsupervised approaches for entity relationships acquisition [Frisoni et al., 2020a, Frisoni and Moro, 2020] and classes extraction Domeniconi et al. [2014b], Moro and Masseroli [2021] to avant-garde semantic parsing solutions such as event extraction [Frisoni et al., 2021]; (ii) retrieval-enhanced techniques [Moro and Valgimigli, 2021]. Lastly, as proposed for communication networks [Lodi et al., 2010, Cerroni et al., 2013], tracking and propagating knowledge refinements across sentences could be critical when tackling extended sequences.

Ethics Statement

Real-world organizations and research teams with a small amount of labeled document summary pairs can get the most out of using our solution. However, because of biases in current pre-trained language models [Liang et al., 2021], humans should supervise the summarization process to guarantee the quality of the generated summaries. For this work, we do not deal with sensible or dangerous data. All datasets and models are publicly available.

3.3
Efficient Memory-Enhanced Transformer for Long-Document Summarization in Low-Resource Regimes

Originally published as: *Gianluca Moro, Luca Ragazzi, Lorenzo Valgimigli, Giacomo Frisoni, Claudio Sartori, and Gustavo Marfia. Efficient memory-enhanced transformer for long-document summarization in low-resource regimes. Sensors, 23(7):3542, 2023d. doi: 10.3390/S23073542. URL https://doi.org/10.3390/s23073542*

Abstract

Long document summarization poses obstacles to current generative transformer-based models because of the broad context to process and understand. Indeed, detecting long-range dependencies is still challenging for today's state-of-the-art solutions, usually requiring model expansion at the cost of an unsustainable demand for computing and memory capacities. This paper introduces Emma, a novel efficient memory-enhanced transformer-based architecture. By segmenting a lengthy input into multiple text fragments, our model stores and compares the current chunk with previous ones, gaining the capability to read and comprehend the entire context over the whole document with a fixed amount of GPU memory. This method enables the model to deal with theoretically infinitely long documents, using less than 18 and 13 GB of memory for training and inference, respectively. We conducted extensive performance analyses and demonstrate that Emma achieved competitive results on two datasets of different domains while consuming significantly less GPU memory than competitors do, even in low-resource settings.

3.3.1 Introduction

In the natural language processing (NLP) field, long document summarization (LDS) synthesizes a lengthy input text while retaining relevant information, a critical task to help experts in analyzing massive documents. State-of-the-art (SOTA) solutions are based on transformers [Vaswani et al., 2017] and struggle to deal with prolonged documents because of the self-attention mechanism that requires high-memory GPUs to address its quadratic memory growth regarding input size. Most documents, such as contracts and research papers, breach endurable input size limits. This issue has recently opened new research directions towards attention approximations with linear complexity [Choromanski et al., 2021, Huang et al., 2021]. Nevertheless, despite their success, efficient transformers are still GPU-demanding and bound to the input size, e.g., 48 GB for 16K source tokens [Beltagy et al., 2020].

A promising approach to mitigate this issue is exploiting memory-based strategies [Dai et al., 2019, Rae et al., 2020]. Specifically, language models are trained to

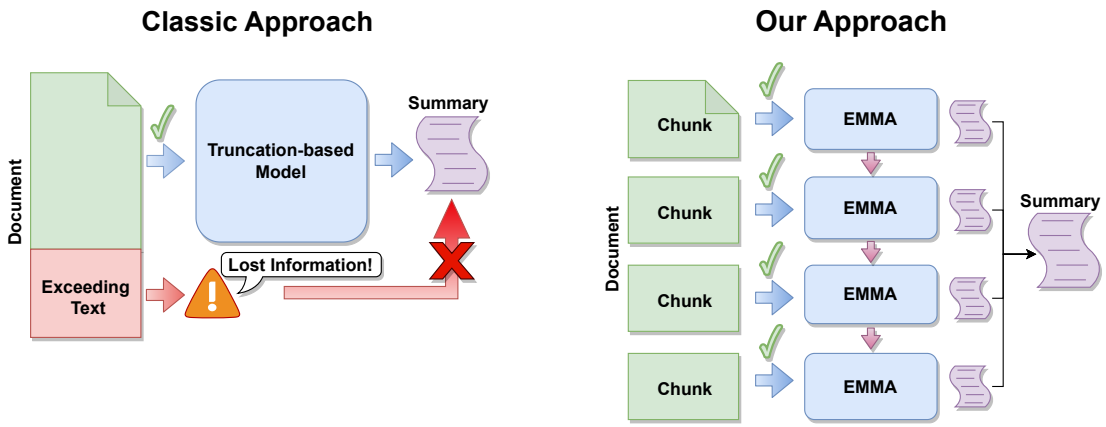


FIGURE 3.6. EMMA overview, our proposed memory-enhanced approach.

recurrently process a chunk-divided input, writing and reading the past latent knowledge at each step; in this way, the GPU is restricted to working with several length-constrained text fragments instead of elaborating the entire source document at once. Current memory-enhanced models are found on encoder- or decoder-only architectures, preventing their application on sequence-to-sequence tasks such as LDS. Indeed, the most promising research directions for abstractive single- and multi-document summarization currently follow an encoder–decoder paradigm, with lightweight models surpassing or holding up against decoder-only summarizers with hundreds of billions of parameters [Floridi and Chiriatti, 2020].

In this work, we present EMMA, an efficient memory-enhanced encoder–decoder language model for LDS. EMMA reads long inputs chunk by chunk (Figure 3.6), saving intermediate knowledge and enriching the current context with previous salient information via *cross-memory attention*. We modified the vanilla transformer with new custom memory layers (short- and long-term memory), decoupling the mutual relationship between GPU need and input size.

We experimented on datasets from different domains, showing EMMA's generality and capacity to summarize long inputs with comparable results to strong baselines despite using significantly less GPU memory at training and inference time.

To sum up, our contributions are the following.

- We introduce EMMA, a novel memory-enhanced encoder–decoder transformer for LDS.
- We perform extensive analyses showing SOTA's performance at low GPU cost, on full-resource summarization (i.e., training on all training samples), and few-shot learning.
- The GPU impact of EMMA remained fixed regardless of input length.

3.3.2 Related Work

3.3.2.1 Transformers

Transformer-based models are the de facto standard in many NLP tasks [Devlin et al., 2019, Liu et al., 2019]. However, their performance is better as parameters increase, leading to the creation of massive models [Floridi and Chiriatti, 2020, Raffel et al., 2020]. Despite their success, current works have had problems in dealing with prolonged input sequences because their core layer, namely, self-attention, scales quadratically with input size. For example, the text supplied to BART must not go beyond 1024 subword tokens, and longer documents have to be cut. Further, most models are pre-trained on sequences of just 512 tokens [Raffel et al., 2020], rendering them unable to handle real-world inputs in downstream tasks. Consequently, meaningful context and details for the summarizers are typically lost. To fill this gap, self-attention has been approximated with linear functions. BIGBIRD [Zaheer et al., 2020] and LONGFORMER [Beltagy et al., 2020] leverage window-based attention. NYSTRÖMFORMER [Xiong et al., 2021] uses Nyström-based matrix decomposition. PERFORMER [Choromanski et al., 2021] relies on kernel methods. With these notable contributions, large language models can read texts up to 16K tokens with a GPU of 48 GB memory [Beltagy et al., 2020]. Regarding architectures, fine-tuned encoder–decoder models are notoriously dominant compared to zero-shot prompting on large decoder-only language models [Goyal et al., 2022a]. Businesses can achieve high summarization quality and versatility with lower costs and more flexibility regarding training and deployment, with networks running locally on private servers and GPUs.

3.3.2.2 Memory-Based Transformers

The link between memory and neural networks was initially explored with differentiable reading and writing operations in the neural Turing machine [Graves et al., 2014, Gülçehre et al., 2018], Differentiable computing networks [Graves et al., 2016], and gated recurrent units [Moro et al., 2018]. However, using memory in the transformer is a less investigated research path. TRASFORMERXL [Dai et al., 2019] was the first to create a recurrent short-term layer-level memory. In contrast, COMPRESSIVE TRANSFORMER [Rae et al., 2020] adds long-term memory to the recurrent one. ERNIE-DOC [Ding et al., 2021] improves the memory flow, letting the model deal with infinitely long sequences. ∞-FORMER [Martins et al., 2022] leverages a continuous attention framework [Martins et al., 2020] to create theoretically infinite memory. Importantly, these models are decoder-only and mainly applied to long-input open-generation tasks, thus neglecting LDS. The latest works also focused on top-k text-retrieval operations from read-only memories with pre-computed embeddings [Borgeaud et al., 2022, Frisoni et al., 2022c]; despite the encouraging performance gain, they rarely support representation updates and have not been tested on document summarization.

3.3.2.3 Long Document Summarization

SOTA LDS solutions utilize different methods to read long sequences. Hierarchical models [Rohde et al., 2021] iteratively merge paragraph-level dependencies. Segmentation-based approaches [Wu et al., 2021a, Moro and Ragazzi, 2022, Zhang et al., 2022d] with fusion-in-decoder [Ivgi et al., 2022] and marginalized decoding [Liu et al., 2022c] divide the input into meaningful units to produce a summary. Extract-then-abstract procedures [Bajaj et al., 2021] pick a subset of relevant sentences from the source to generate the outline, eventually relying on marginalization [Mao et al., 2022, Moro et al., 2022a]. Lastly, efficient transformers with sparse attention layers [Beltagy et al., 2020, Huang et al., 2021, Tay et al., 2023] read greater input than quadratic ones do while not fully leveraging the original self-attention mechanism.

3.3.3 Background

LDS tasks compress a long input text into a coherent short summary. Given the extensive and successful use of the transformer architecture, a document is long if its number of tokens poses processing complications to standard language models. Even if a formal definition does not exist, texts comprising > 1024 tokens are commonly "long." This threshold is also the maximal input size for well-known quadratic models such as BART [Lewis et al., 2020a] and PEGASUS [Zhang et al., 2020a].

The problem of LDS can be formalized with an input document $\mathcal{X}$ and its target summary $\mathcal{Y}$. Since a classical transformer needs to rely on input truncation, memory can help in preserving salient information. Intuitively, we can split a long input into chunks $\{c_1, c_2, \ldots, c_n\}$ and give them one by one to a model that could (i) read each chunk, (ii) save the relevant information in the memory and reuse it for subsequent chunks, and (iii) generate a summary for each chunk. Eventually, the final summary is obtained by concatenating chunk-level summaries.

Unfortunately, existing memory-based transformers are limited to $(\mathcal{X}, \mathcal{Y})$ tasks with a target for each input text. This setting is a substantial limitation and the main reason why memory-based transformers have not yet been applied to LDS where there is a single target even after segmentation.

3.3.4 Method

EMMA is a novel efficient memory-augmented transformer for LDS. Our model relies on a text segmentation algorithm and memory layers to recurrently read the provided input, chunk after chunk; at each step, it stores the relevant information and compares it with previous information. EMMA can handle infinitely long documents with a fixed amount of GPU memory.

3.3.4.1 Text Segmentation

Let $\mathcal{X} = \{x_1, \ldots, x_x\}$ and $\mathcal{Y} = \{y_1, \ldots, y_y\}$ be the long input document and related target summary, respectively, where each $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ is a sentence. We seg-

Algorithm 5 Text Segmentation

Input: $\mathcal{X} = \{x_1, \ldots, x_x\}$ ▷ Input sentences
Parameters: L_c ▷ Number of tokens per chunk
Output: $\mathcal{C}$ ▷ Set of chunks

$\mathcal{C} \leftarrow \emptyset$
$c \leftarrow \emptyset$
for $x_i \in \mathcal{X}$ **do**
  $l \leftarrow len(c) + len(x_i)$
  if $l < L_c$ **then**
    $c \leftarrow c \circ x_i$
  else
    $\mathcal{C} \leftarrow \mathcal{C} + c$
    $c \leftarrow \emptyset$
  end if
end for
if $len(c) \neq \emptyset$ **then**
  $\mathcal{C} \leftarrow \mathcal{C} + c$
end if
return $\mathcal{C}$

mented $\mathcal{X}$ into non-overlapping chunks $\mathcal{C}$ of max L_c tokens with a sentence-level segmentation algorithm (Algorithm 5). We started with an empty chunk c and iteratively added sentences until L_c. After constructing the chunks, we paired each target summary sentence with the chunk that maximized the ROUGE-1 precision metric [Moro and Ragazzi, 2022], deriving small source-target pairs. Consequently, we turn the problem from $\{(c_1, c_2, \ldots, c_n), \mathcal{Y}\}$ to $\{(c_1, t_1), (c_2, t_2), \ldots, (c_n, t_n)\}$, where $c_1 \circ c_2 \circ \cdots \circ c_n = \mathcal{X}$ and $t_1 \circ t_2 \circ \cdots \circ t_n = \mathcal{Y}$, with $\circ$ denoting string concatenation.

3.3.4.2 Model Architecture

We enhanced the transformer-based model BART [Lewis et al., 2020a] with a recurrent layer-level memory where the model stores past information. Specifically, we allowed for the model to compare current chunk c_i with information related to previous ones $\{c_1, \ldots, c_{i-1}\}$. The original layers of the BART encoder are composed of self-attention and feed-forward blocks with residual connections. As shown in Figure 3.7, we added a layer-level memory and a second attention block, termed *cross-memory attention*, to perform reading and writing operations. The memory is a single matrix M.

Cross-Memory Attention We added cross-memory attention after a residual connection that follows the self-attention of the classical BART encoder layer. At the i-th step, this module enables the model to juxtapose the hidden states h_i of chunk c_i with $(h_1, \ldots, h_{i-1})$ via cross-attention. Around this layer, we added a residual connection to let the model learn how much to use the memory. Formally, hidden state h_i^m is acquired

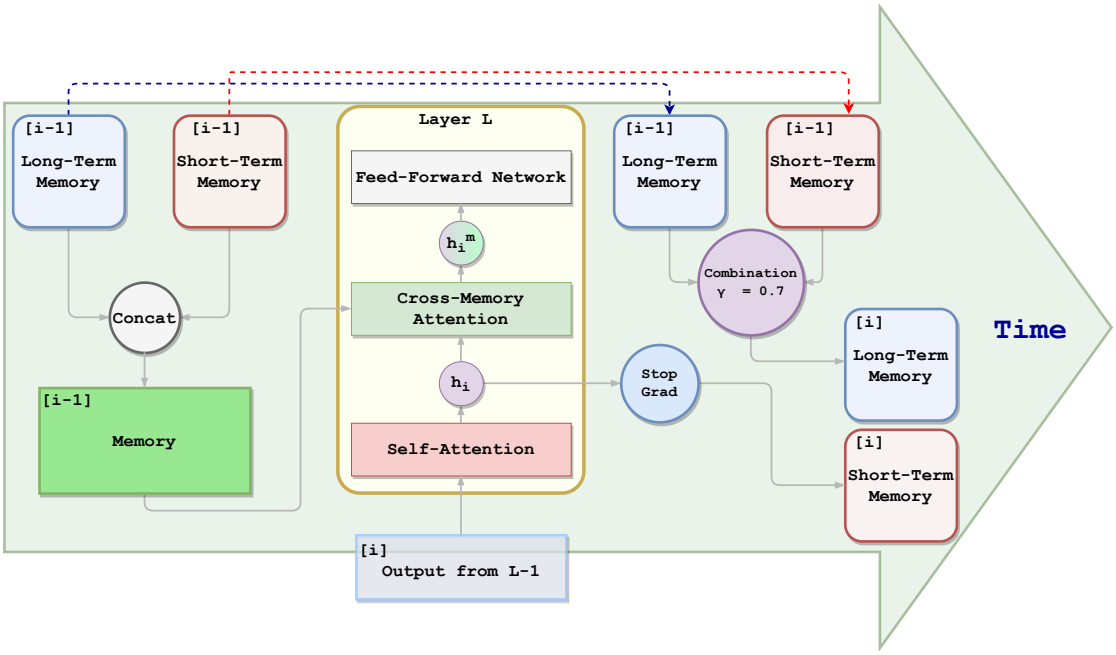


FIGURE 3.7. Graphical representation of our proposed memory-enhanced layer. (left to right) Long- and short-term memories are (i) concatenated, (ii) fused with input hidden representation via cross-memory attention, and (iii) the input hidden representation becomes the new short-term memory while the long-term memory is updated with information from the previous short-term one. To simplify the figure, we do not depict the residual connections.

with the following formula:

$$h_i^m = \mathcal{N}(h_i + \mathcal{C}(h_i, \mathsf{M}_{i-1})), \tag{3.11}$$

where $\mathcal{N}$ is a normalization layer, $\mathcal{C}$ is the cross-memory attention layer, M_{i-1} is the memory, and h_i is the hidden state after the self-attention.

Memory Writing We equipped each layer with a memory to store helpful information for the next step, overriding the previous memory. After performing cross-memory attention for the i-th chunk and generating h_i^m, h_i is given to the memory module. In detail, h_i passes through a stop gradient function, $SG(h_i)$, and becomes the new memory matrix:

$$\mathsf{M}_i = SG(h_i). \tag{3.12}$$

By stopping the gradient, the GPU memory used at training does not increase with the number of chunks, allowing for the model to work with a theoretically infinitely size document.

Long-Term Memory With the memory overridden at each step, we may lose long-term details. For this reason, we improved the architecture by adding a long-term

memory. In particular, we moved M_{i-1} into a different matrix M_i^l, which we call the long-term memory matrix, before overriding it with the new hidden state h_i. Memory M_{i-1} was compressed and combined with the long-term memory matrix M_{i-1}^l as follows:

$$\mathsf{M}_i^l = (1-\gamma)\cdot\mathsf{M}_{i-1}^l + \gamma\cdot\mathsf{M}_{i-1}, \tag{3.13}$$

where γ is a compress ratio empirically set to 0.7. The final memory M_i^c used for the cross-memory attention was obtained by concatenating the short- and long-term memories:

$$\mathsf{M}_i^c = \mathcal{C}(\mathsf{M}_{i-1}, \mathsf{M}_{i-1}^l), \tag{3.14}$$

where $\mathcal{C}$ is the concatenation function.

3.3.4.3 Training and Inference

EMMA takes as input the chunk–target pairs and was trained to generate the next output token for each target by minimizing the negative log-likelihood:

$$\mathcal{L} = -\frac{1}{|t|}\sum_{i=1}^{|t|}\log p(y_i|y_{1:i-1}, c), \tag{3.15}$$

where c is the input chunk, and $y_{1:t}$ are the tokens from position 1 to t of its target t. For the training process, we took only the chunk–target pairs (c_i, t_i), such that $t_i \neq \emptyset$. Instead, at inference time, we considered all the chunks and concatenated the chunk-level summaries to establish the final prediction.

3.3.4.4 Space Complexity

Our model, EMMA, has quadratic space complexity regarding the length of the input chunks. Given a predefined max chunk size L_c, a document with size L_D is split at most into $\lceil \frac{L_D}{L_c} \rceil$ chunks. Thanks to our solution, the chunks are individually processed and synthesized, and their summaries are concatenated to produce the final output (Figure 3.6). Hence, the space complexity to summarize the entire input document is $\mathcal{O}(L_c^2)$; since it relies on the model's encoder self-attention for a single chunk, it remains fixed regardless of the document length. As our model was built upon BART, the encoder self-attention had quadratic complexity in the chunk size.

3.3.5 Experiments

3.3.5.1 Evaluation Datasets and Training Settings

We tested EMMA under (i) full training and (ii) few-shot learning scenarios by utilizing datasets containing long documents on different specific domains. In (i), we took GOVREPORT [Huang et al., 2021] and PUBMED [Cohan et al., 2018] as the evaluation benchmarks. GOVREPORT collects reports from government research agencies, while

TABLE 3.15. Statistics of the LDS datasets used as evaluation benchmarks. (top) Full training; (bottom) few-shot learning.

Dataset	Samples	Source *#avg Words*	Target *#avg Words*
GOVREPORT	19,466	9409.4	553.4
PUBMED	133,215	3224.4	214.4
BILLSUM	23,455	1813	207.7

PUBMED comprises biomedical research articles. In (ii), we worked with BILLSUM [Kornilova and Eidelman, 2019], which consists of U.S. congressional bills. Statistics of the datasets are reported in Table 3.15. To reduce the training time and energy consumption, we used a maximum of 20 K training instances for each dataset. For GOVREPORT, we used the default training and test splits: the training set comprised 17,517 instances, and the test set contained 973 examples. For PUBMED, we used the first 20,000 samples of the training set and the full test set of 6658 instances. For BILLSUM, following prior works [Zhang et al., 2020a, Chen and Shuai, 2021], we utilized the first 10 and 100 training instances (the same sampling strategy as that for validation).

We adopted the ROUGE-1/2/L standard [Lin, 2004] as the automatic LDS metric. Inspired by Moro et al. [2023b], we also computed $\mathcal{R} = {}^{\text{avg}(r_1,r_2,r_L)}/_{1+\sigma_r^2}$, where σ_r^2 is the ROUGE F1 score variance. In this way, we derived an aggregated judgment that, in the case of equal $r_{1/2/L}$ averages, penalizes generations with heterogeneous results across dimensions. To contain the variance effect that was only designed to slightly refine average values, we considered $r_{1/2/L} \in [0, 1]$ and $\mathcal{R} \in [0, 1]$ (the higher, the better). Lastly, we performed qualitative analysis to complement the notorious lexical superficiality of ROUGE [Frisoni et al., 2022a].

3.3.5.2 Baselines

- *Full training.* To understand the contribution of our new memory, we examined BART [Lewis et al., 2020a], the skeleton model that we had extended. Then, we contemplated SOTA models on BART that do not perform any further pre-training, like ours. We chose LED [Beltagy et al., 2020] and HEPOS [Huang et al., 2021], which leverage various efficient attention mechanisms and are capable of reading the entire long input. In particular, in HEPOS, we considered locality-sensitive hashing (lsh) and sinkhorn. We lastly evaluated our model against SUMM^N [Zhang et al., 2022d], a segmentation based solution.

- *Few-shot learning.* We compared it with well-known low-resource abstractive summarizers. PEGASUS [Zhang et al., 2020a] is a transformer-based model with a summarization-specific pre-training objective that allows for fast adaption through a few labeled samples. MTL-ABS [Chen and Shuai, 2021] combines transfer learning and meta-learning from multiple corpora by using adapter modules as bridges. To judge the contribution of document segmentation versus memory, we contrasted

Emma with Se3 [Moro and Ragazzi, 2022], a semantic self-segmentation approach for LDS under low-resource regimes with proven strength in data scarcity conditions. Similarly to our model, Se3 avoids truncation by creating highly correlated source–target chunk-level pairs with lengths modulated to fit into the GPU memory. Despite empowering the chunk definition process with deep metric learning following information retrieval techniques [Moro and Valgimigli, 2021, Meng et al., 2022b, Moro et al., 2022b, Moro and Salvatori, 2022], Se3 represents a general pre-processing technique for any transformer where chunks are individually summarized and then concatenated (no memory extension or architectural changes). To ensure fairness, we refer to Se3+Bart.

3.3.5.3 Experimental Settings

We trained Emma for 10 epochs in two versions, the base (192 M trainable parameters) and large (508 M trainable parameters). We report the results of the best-performing checkpoint on the validation set. We used the AdamW optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$, and set the dropout to 10%. The learning rate was 3×10^{-5}, the batch size was 1, and the seed was fixed to 42 for reproducibility. At inference time, we set the beam width to 5 for all experiments and prevented the repetition of n-grams of size 5. We used a summary length between 400 and 1000 for GovReport, and 100 and 700 for PubMed with the repetition penalty set to 1. We conducted the work on a workstation using a single GPU RTX 3090 with 24 GB dedicated graphics memory, 64 GB RAM, and an AMD EPYC 7443 24-core processor. The operative system was Ubuntu 20.04.3 LTS; the development environment was a docker container with an official Hugging Face image (`huggingface/transformers-pytorch-latest-gpu`, accessed on 23 March 2023). We implemented the code using Python 3.8, PyTorch to handle gradient optimization, and Hugging Face for the neural models (`https://huggingface.co/models`, accessed on 23 March 2023) and datasets (`https://huggingface.co/datasets`, accessed on 23 March 2023).

3.3.5.4 Performance Evaluation

We extensively measured Emma's performance quantitatively and qualitatively. All ROUGE scores detailed in this section are expressed as percentages.

Full-Training Results Table 3.16 reports the LDS results under full-training settings. Compared to traditional SOTA encoder–decoder summarizers without memory, Emma achieved competitive or higher ROUGE F1 scores, with significant improvements in hardware requirements (see Section 3.3.5.5). The outcomes show that Emma captures salient information if either equally distributed in the long input (GovReport) or accumulated in the first partitions of documents (PubMed).

Few-Shot Learning By supervising our model on limited data, we analyze how quickly Emma leverages the inner pre-trained model. Results in Table 3.17 indicate that

TABLE 3.16. Full-training ROUGE F1 scores on GOVREPORT and PUBMED. Baseline results are from the original papers. Bold and underline denote the best and second-best scores.

Model	GovReport R1/R2/RL	$\mathcal{R}$	PubMed R1/R2/RL	$\mathcal{R}$	Average $\mathcal{R}$
Baselines					
BART [Lewis et al., 2020a]	52.83/20.50/50.14	40.29	45.36/18.74/40.26	34.33	37.31
HEPOS-LSH [Huang et al., 2021]	55.00/21.13/51.67	41.63	**48.12**/**21.06**/42.72	**36.80**	39.99
HEPOS-SINKHORN [Huang et al., 2021]	56.86/22.62/53.82	43.39	47.96/20.78/42.53	36.59	39.99
LED [Beltagy et al., 2020]	**59.42**/**26.53**/**56.63**	**46.50**	47.00/20.20/42.90	36.20	**41.35**
SUMM^N [Zhang et al., 2022d]	56.77/23.25/53.90	43.64	–	–	–
Ours					
EMMA-BASE	58.78/24.30/55.29	45.04	44.31/17.35/40.91	33.70	39.37
EMMA-LARGE	59.39/25.27/55.90	45.77	46.70/19.51/**43.42**	36.01	40.89

EMMA outperforms previous summarizers, revealing its learning skills in low-resource. Higher ROUGE scores over SE3 corroborate the memory value more than segmentation only does.

Ablation Studies To assess the importance of our architecture's main components, we performed a set of ablation studies (Tables 3.18 and 3.19), using the GOVREPORT training settings with 1000 samples for 3 epochs. In particular, we investigated the following design choices.

- *w/ Backprop*: We attempted not to stop the backpropagation within the current chunk but allowed it to go back in time to previous steps. Results show a performance drop, probably due to the increased learning complexity. This approach is unexplored in memory-enhanced transformers and deserves greater research attention.
- *w/ Long-term memory*: we removed the long-term memory module. Results worsened, ascertaining the contribution of this component to the final summary quality.
- *Memory layers*: We performed a series of experiments to determine which layers turned the memory on. The last two were the best ones, aligned with Rae and

TABLE 3.17. Few-shot learning ROUGE F1 scores on the BILLSUM dataset using 10 and 100 training instances. Baseline results are from the original papers. Bold and underline denote the best and second-best scores.

Model	BillSum (10) R1/R2/RL	$\mathcal{R}$	BillSum (100) R1/R2/RL	$\mathcal{R}$	Average $\mathcal{R}$
Baselines					
PEGASUS [Zhang et al., 2020a]	40.48/18.49/27.27	28.52	44.78/26.40/**34.40**	34.99	31.76
MTL-ABS [Chen and Shuai, 2021]	41.22/18.61/26.33	28.47	45.29/22.74/29.56	32.24	30.36
SE3 [Moro and Ragazzi, 2022]	46.58/22.03/28.23	31.93	49.88/26.84/33.33	36.34	34.14
Ours					
EMMA-BASE	**46.77**/**22.95**/**28.81**	**32.51**	**50.78**/**28.58**/34.27	**37.55**	**35.03**

TABLE 3.18. Ablation studies to validate the components of the solution. The best results are in bold.

Model	GovReport R1	GovReport R2	GovReport RL
Full	**59.99**	**23.96**	**56.35**
w/Backprop	41.44	12.66	39.98
w/o Long-term memory	58.83	22.61	55.03

Razavi [2020], where the authors claimed that TRANSFORMERXL operated better with memory only on layers in the second half of the encoder.

3.3.5.5 Analysis of the GPU Impact

GPU Memory Usage One of the main benefits of adopting memory components into language models is that the GPU memory consumption rate remains fixed regardless of the input document length. SOTA solutions with efficient attention mechanisms, such as LED and HEPOS, have a maximal limit on the number of tokens that they can read simultaneously. Therefore, applying such models to domains characterized by extremely long sources (e.g., books, meeting dialogues, trials) is hard if not impossible. Memory can precisely mitigate this problem: at inference time, theoretical GPU usage depends only on the dimension of the model. This property held for our solution, even during training, thanks to interrupting the backpropagation through chunks. Figure 3.8 qualitatively exhibits the training time of GPU utilization for 10 artificially crafted documents ordered by length. We compared our model with the best-performing linear attention transformers, namely, LED-base and LED-large (retrained by us). EMMA's GPU need was stable for all documents despite the increase in source tokens.

In linear-attention-based solutions such as LED, memory usage scales linearly regarding input length. However, these models still suffer from serious scalability issues that preclude their application. For example, according to their original papers [Beltagy et al., 2020, Huang et al., 2021], both HEPOS (batch size 2) and LED (batch size 1) require 48 GB of GPU memory to fit and train the models for processing 16 K input tokens. Moreover, their functions for approximating quadratic self-attention perform slightly

TABLE 3.19. Ablation study to assess which layer turns on the memory. The best results are in bold.

Memory-Layer	GovReport R1	GovReport R2	GovReport RL
All	58.71	23.18	55.73
Last three	59.22	**24.10**	55.96
Last two	**59.99**	23.96	**56.35**
Last one	58.76	22.91	55.51

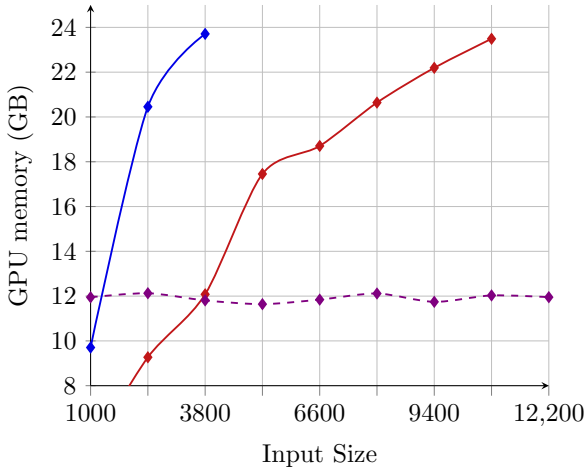


FIGURE 3.8. GPU memory occupation at training time by varying input size (`batch_size=1`).

worse with short inputs. Similarly, DYLE uses 48 GB with batch size 8. Its memory usage depends on the number of top-K snippets to select from the input source. In our 24 GB hardware configuration, only $K = 10$ was manageable corresponding to F1 ROUGE scores equal to 54.98/24.10/51.25 [Mao et al., 2022], which were significantly worse than those of EMMA in the same settings.

Our solution achieves comparable results on GOVREPORT using less than 24 GB of GPU memory. Similar to DYLE, our GPU memory consumption did not scale with the document length, but the minimal amount required was significantly lower.

Chunk Size Analysis We split the input document into chunks; the memory used at inference time only depended on the one needed to process a single chunk. Table 3.20 shows how the performance changed by varying the chunk length. Since we segmented the input document, the memory used at the inference time only depended on the one needed to process a single chunk. Table 3.20 depicts how summarization effectiveness changed by varying the chunk size bounds. ROUGE scores slightly worsened by decreasing the number of tokens per chunk, but our model powerfully maintained a good trade-off between chunk size and summary quality. Leveraging past information thanks to memory is vital for generating high-quality summaries, especially when decreasing the chunk size (i.e., increasing the number of chunks). In a nutshell, EMMA achieves highly competitive summarization performance even with reduced chunk sizes, which implies downsized GPU memory demand. Figure 3.9 shows the impact of the chunk size from an efficiency perspective, measuring the GPU usage and memory occupation. A significant GPU memory drop appeared with a chunk size between 384 and 512 tokens. Further, the GPU usage scaled linearly with the chunk size. Chunks between 384 and 512 tokens had the best trade-offs. These outcomes show that memory can be central in low-resource models, uncoupling the GPU impact and the input length.

TABLE 3.20. Results of over 100 documents of the test set from GOVREPORT while changing chunk size.

Chunk Size	R1	R2	RL
256–384	58.43	23.44	54.31
384–512	60.92	25.21	56.93
512–640	61.65	26.50	58.12
640–768	61.46	26.15	58.21

3.3.5.6 Human Evaluation

We conducted a comprehensive human evaluation study to better gauge the quality of the summaries produced by EMMA regarding LED (the main full-training competitor according to Table 3.16). We randomly selected 50 document–summary pairs from the test sets of GOVREPORT and PUBMED (25 from each source). We asked three evaluators who were proficient in English with legal and medical competencies to select their most and least preferred predictions according to informativeness, fluency, factuality, and succinctness, i.e., best–worst scaling [Louviere and Woodworth, 1991, Louviere et al., 2015]. We randomized the order of summaries within pairs to guard the rating against being gamed. Our setup with human instructions is illustrated in Figure 3.11. The annotation process took approximately 6 h per judge, 18 h in total. The average Kendall coefficient among all evaluators' inter-rater agreement was 0.60. All evaluation files were publicly released for transparency and reproducibility: `https://github.com/disi-unibo-nlp/emma` (accessed on 23 March 2023). Results are outlined in Table 3.21, showing the overall percentage of times that a particular system was the most preferred summary source. Additionally, we plot the distribution of dimension-specific votes in Figure 3.10. Across both quality dimensions and datasets, we observed a clear preference for EMMA. LED tended to be less abstractive and to have more extended outputs, often cut before reaching the `end-of-sentence` token, focused on the first part of the document. Instead, EMMA was much more concise, going straight to the point and covering all the relevant content mentioned in the document with high frequency and factuality. The overall advantage of our solution is strongly accentuated as

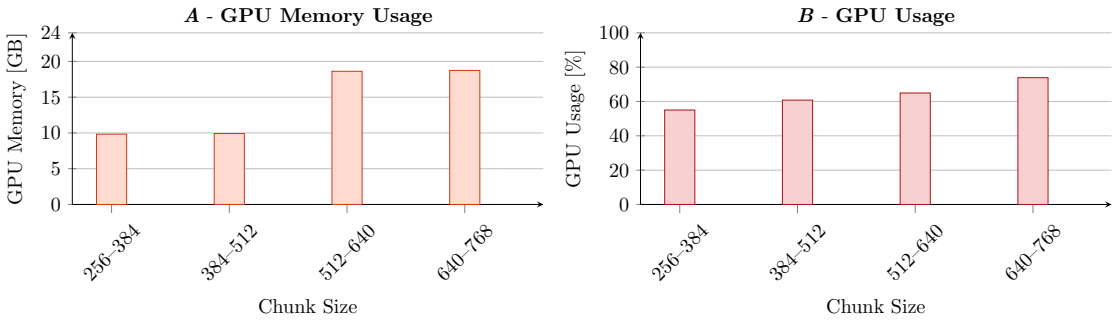


FIGURE 3.9. Graphs on how the chunk size hyperparameter of EMMA impacts the GPU. (**A**) GPU memory usage (0–24 GB); (**B**) GPU computational power usage (0–100%).

TABLE 3.21. Percentage of times that a summarizer was selected as the best from all evaluators. Annotators preferred EMMA outputs over LED for approximately 70% of the sampled document–summary pairs. The best results are on a green background.

Model	GovReport	PubMed	Overall
LED	22.67	42.33	32.50
EMMA	77.33	57.67	67.50

the length of the target summary increased (GOVREPORT summaries were, on average, 2.58× longer than those of PUBMED).

3.3.6 Conclusions

Although augmenting transformers with memory is receiving less attention and effort than efficient transformers, it can play a pivotal role in low-resource settings and domains with extremely long documents. In this work, we presented EMMA, the first memory-enhanced encoder–decoder transformer for long-document summarization. The proposed architecture leverages two fundamental elements: (i) a segmentation algorithm for splitting the input document into chunks and pairing them with the most related parts of the target summary, and (ii) a recursive memory module capable of storing information from past chunks. We tested our solution with multiple datasets of different domains, obtaining competitive results with state-of-the-art models under full-training conditions and outperforming prior works in few-shot learning. Exceptionally, depending only on the chunk size, the GPU need remained constant regardless of the whole document length. Compared to segmentation-only techniques, our memory component boosted summarization quality, avoiding treating each chunk independently, and better exploiting their semantic linkage. We also verified that the chunk size could be kept low without significant drops in summarization results, enabling SOTA performance on limited hardware. In-depth ablation studies support our architectural design choices. This study promotes novel research toward efficient memory-enhanced language models.

3.3.7 Limitations and Future Directions

Our document segmentation algorithm requires the length of the golden summary to not be too short; otherwise, paired targets are composed of a few tokens. According to our

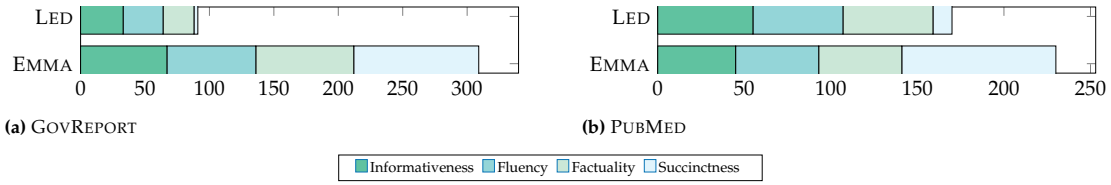


FIGURE 3.10. Distribution of votes per quality dimension (cumulative best-selection percentages).

TABLE 3.22. List of the models and datasets used in this study. All the links have been last accessed on 23 March 2023.

Model	URL
LED-BASE	https://huggingface.co/allenai/led-base-16384
LED-LARGE	https://huggingface.co/allenai/led-large-16384
BART-BASE	https://huggingface.co/facebook/bart-base
BART-LARGE	https://huggingface.co/facebook/bart-large
Dataset	**URL**
GOVREPORT	https://huggingface.co/datasets/ccdv/govreport-summarization
PUBMED	https://huggingface.co/datasets/ccdv/pubmed-summarization
BILLSUM	https://huggingface.co/datasets/billsum

empirical tests in the ablation studies, enabling backpropagation through chunks led to worse results. Deeper investigations are needed to solve this issue.

Future works should explore memory writing/reading operations with structured information extracted from text, comparing unsupervised techniques for document metadata acquisition (e.g., classes [Domeniconi et al., 2015a, 2017] and entity relationships [Frisoni et al., 2020a, Frisoni and Moro, 2020]) with advanced semantic parsing solutions such as event extraction [Frisoni et al., 2021, 2022d] and abstract meaning representation, which was recently used for knowledge injection into deep neural networks [Frisoni et al., 2023, 2022b]. The community should envisage novel graph representation learning methods [Chen et al., 2022a, Cao et al., 2022a, Ferrari et al., 2022, Frisoni et al., 2022e] to

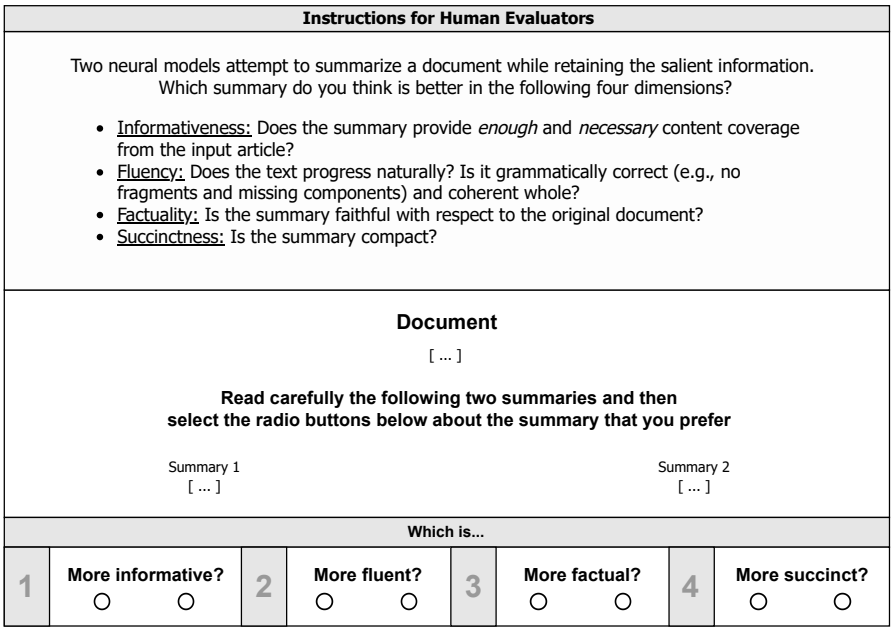


FIGURE 3.11. Human assessment interface.

densely represent multi-relational structured data following a Linked Open Data vision centered on the integration of several source knowledge graphs or relational databases via automatic entity matching [Singh et al., 2017]. Taking inspiration from biology [Domeniconi et al., 2016a, Moro and Masseroli, 2021] and communication networks [Monti and Moro, 2008, Lodi et al., 2010, Moro and Monti, 2012, Cerroni et al., 2013], we underline the importance of managing dynamic scenarios, tracking knowledge refinements among sentences, and propagating information, which is pivotal when processing lengthy inputs. Segmentation strategies and memory-enhanced encoder–decoder transformers could be inspected in other downstream tasks with long documents and cross-dependencies among chunks, such as claim verification with evidence retrieval [Kryscinski et al., 2020, Saeed et al., 2022].

3.3.A References to Models and Datasets

Table 3.22 enumerates all the trained models and datasets used in this study, linking to specific HuggingFace versions.

3.3.B Human Evaluation Insights

The interface with human evaluation instructions is sketched in Figure 3.11.

Chapter 4

Retrieval-Augmented Generation

4.1
Discriminative Marginalized Probabilistic Neural Method for Multi-Document Summarization of Medical Literature

Originally published as: *Gianluca Moro, Luca Ragazzi, Lorenzo Valgimigli, and Davide Freddi. Discriminative marginalized probabilistic neural method for multi-document summarization of medical literature. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 180–189. Association for Computational Linguistics, 2022a. doi: 10.18653/V1/2022.ACL-LONG.15. URL https://doi.org/10.18653/v1/2022.acl-long.15*

Abstract

Although current state-of-the-art transformer-based solutions succeeded in a wide range for single-document NLP tasks, they still struggle to address multi-input tasks such as multi-document summarization. Many solutions truncate the inputs, thus ignoring potential summary-relevant contents, which is unacceptable in the medical domain where each information can be vital. Others leverage linear model approximations to apply multi-input concatenation, worsening the results because all information is considered, even if it is conflicting or noisy with respect to a shared background. Despite the importance and social impact of medicine, there are no ad-hoc solutions for multi-document summarization. For this reason, we propose a novel discriminative marginalized probabilistic method (DAMEN) trained to discriminate critical information from a cluster of topic-related medical documents and generate a multi-document summary via token probability marginalization. Results prove we outperform the previous state-of-the-art on a biomedical dataset for multi-document summarization of systematic literature reviews. Moreover, we perform extensive ablation studies to motivate the design choices

and prove the importance of each module of our method.[1]

4.1.1 Introduction

The task of multi-document summarization aims to generate a compact and informative summary from a cluster of topic-related documents, which represents a very challenging natural language processing (NLP) application due to the presence of redundant and sometimes conflicting information among documents [Radev, 2000]. In the medical domain, in which machine learning plays an increasingly significant role [Domeniconi et al., 2014a, di Lena et al., 2015], multi-document summarization finds application in the generation of systematic literature reviews, a biomedical paper that summarizes results across many studies [Khan et al., 2003]. DeYoung et al. [2021] are the first that address this task, showing the related issues.

State-of-the-art approaches leverage two leading solutions: hierarchical networks that capture cross-document relations via graph encodings [Wan and Yang, 2006, Liao et al., 2018, Li et al., 2020, Pasunuru et al., 2021] or hidden states aggregation [Fabbri et al., 2019, Liu and Lapata, 2019a, Jin et al., 2020], and long-range neural models that apply multi-input concatenation [Xiao et al., 2022]. While effective, these solutions struggle to process clusters of many topic-related documents in low computational resource scenarios [Moro and Ragazzi, 2022] because they need to truncate the inputs. Moreover, pre-trained state-of-the-art transformers are not leveraged despite showing strong performance when fine-tuned in downstream tasks such as single-document summarization [Liu and Lapata, 2019b, Lewis et al., 2020a, Beltagy et al., 2020, Raffel et al., 2020, Zaheer et al., 2020].

Multi-document summarization requires models to have more robust capabilities for analyzing the cluster to discriminate the correct information from noise and merge it consistently. In this work, we propose a discriminative marginalized probabilistic neural method (Damen) that selects worthy documents in the cluster with respect to a shared background and generates the summary via token probability marginalization.

The marginalization of the probability has been successfully applied in past NLP models such as pLSA [Hofmann, 1999] to learn the word probability distribution in documents by maximizing the likelihood. Recently, new deep neural models that use probability marginalization approaches have been proposed for the question-answering task, where, however, each input/output sentence is generally several orders of magnitude shorter than sets of documents in multi-document summarization [Guu et al., 2020, Lewis et al., 2020b].

To the best of our knowledge, we are the first that propose such a method for multi-document summarization. To this aim, we conduct experiments on the only medical dataset for multi-document summarization of systematic literature reviews (Ms2). Besides, we perform extensive ablation studies to motivate the design choices and prove the importance of each component of our method.

To sum up, our contributions are as follows:

[1] The solution of this paper is available at https://disi-unibo-nlp.github.io/projects/damen

- We propose a novel probabilistic neural method for multi-document summarization (DAMEN) that discriminates the summary-relevant information from a cluster of topic-related documents and generates a final summary via token probability marginalization.
- We advance the research in the medical domain, experimenting with a biomedical multi-document summarization dataset about the generation of systematic literature reviews.
- We show that our solution outperforms previous state-of-the-art solutions, achieving better ROUGE scores. Furthermore, we extensively prove the contribution of each module of our method with ablation studies.

4.1.2 Related Work

We describe related works on multi-document summarization categorized on model architectures.

Flat Solutions Flat concatenation is a simple yet powerful solution because the generation of the multi-document summary is treated as a single-document summarization task, thus it can leverage state-of-the-art pre-trained summarization models. Consequently, processing all documents as a flat input requires models capable of handling long sequences. As previously experimented by DeYoung et al. [2021], Xiao et al. [2022] proposed to leverage the Longformer-Encoder-Decoder model [Beltagy et al., 2020] pre-trained with a novel multi-document summarization specific task. They proved that a long-range transformer that encodes all documents is a straightforward yet effective solution, and they achieved new state-of-the-art results in several multi-document summarization datasets. However, such models may struggle to handle a massive cluster of topic-related documents since they need to truncate them because of architectural limits. Further, processing all documents in a cluster could be noisy if some of them are not relevant or factual with respect to the summary.

Hierarchical Solutions To better preserve cross-document relations and obtain semantic rich representations, hierarchical concatenation solutions leverage graph-based techniques to work from word and sentence-level [Wan and Yang, 2006, Liao et al., 2018, Nayeem et al., 2018, Antognini and Faltings, 2019, Li et al., 2020] to document-level [Amplayo and Lapata, 2021]. Other hierarchical approaches include multi-head pooling and inter-paragraph attention architectures [Liu and Lapata, 2019a], attention models with maximal marginal relevance [Fabbri et al., 2019], and attention across different granularity representations [Jin et al., 2020]. Such models are often dataset-specific because of the custom architecture, so they struggle to adapt to other datasets and effectively leverage pre-trained state-of-the-art transformers.

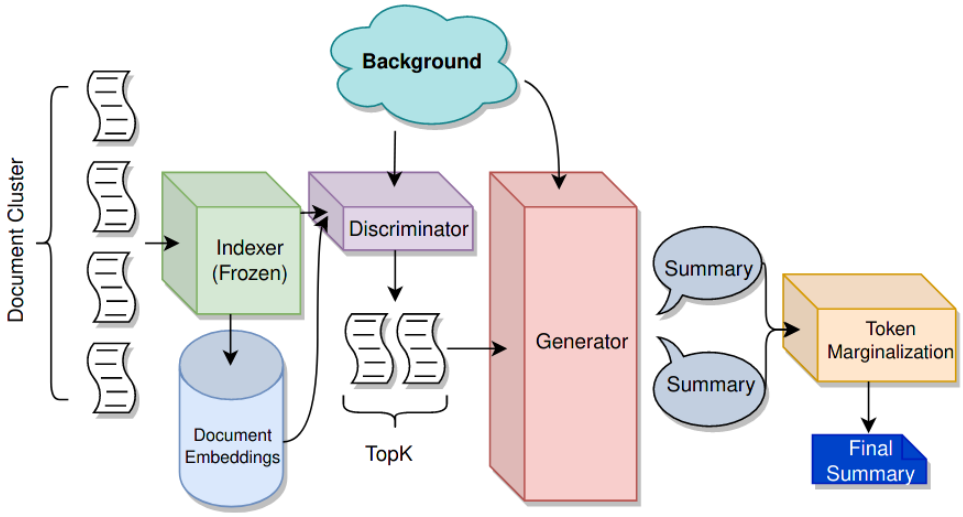


FIGURE 4.1. The overview of DAMEN, our probabilistic neural method for multi-document summarization of medical literature. First, two BERT models (i.e., *Indexer* and *Discriminator*) encode the background and documents in the cluster, creating dense embedding representations. Then, the *Discriminator* selects the top K documents via inner product with the background. Afterward, the background is concatenated with each document retrieved and the new textual inputs are given to BART to generate the multi-document summary by marginalizing the token probability distribution at decoding time.

Our Solution In this work, we show how the summary-relevant information can be discriminated from a cluster of medical documents by a probabilistic neural method trained end-to-end. In detail, our solution fully leverages pre-trained state-of-the-art transformers without applying input truncation that causes performance drop and discards important contents, unacceptable for a high-social impact domain such as the medical one.

4.1.3 Method

We introduce DAMEN, a discriminative marginalized probabilistic neural method for the multi-document summarization of medical literature based on three components:

- ***Indexer***: it is a neural language model based on BERT architecture [Cohan et al., 2020] that creates a dense representation of documents in the cluster, according to the best practices for information retrieval systems.
- ***Discriminator***: it leverages a BERT model to create the background embedding, which is used to compute a distance score between the embedding of each document in the cluster in order to select the top K ones.
- ***Generator***: it uses a BART model [Lewis et al., 2020a] to produce the final summary via token probability marginalization from the top K documents combined with the background.

While the *Indexer* is a frozen pre-trained model based on BERT, the *Discriminator* and *Generator* are trained end-to-end during the learning phase (Figure 4.1). The overall task can be mathematically formalized as follows. The training tuple is composed of three elements $(y_i, x_i, \mathbf{C}_i)$, where y_i is the ground-truth target summary, $\mathbf{C}_i$ is the cluster of documents used to generate the multi-document summary, and x_i is the background, which is a textual context shared by all $c_j \in \mathbf{C}_i$ used as input of the method, similar to the query in the query-focused multi-document summarization [Su et al., 2020]. The whole pipeline is trained end-to-end to maximize the conditional probability of generating y_i from x_i and $\mathbf{C}_i$ through gradient descend:

$$p(y_i|x_i, \mathbf{C}_i) \tag{4.1}$$

4.1.3.1 Indexer

In this phase, we index each document in the cluster with an embedding generated by a BERT-based model. Such a pre-trained language model is the state-of-the-art in semantic modeling from textual data thanks to the vast knowledge learned during pre-training [Chen et al., 2019], achieving groundbreaking results across an extensive range of NLP downstream tasks even without fine-tuning. For this reason, we use it to create a dense latent representation of each document, called document embedding, which is a vector of continuous numbers that indicates a point in a latent semantic space. The technique we use is known as dense passage retriever (DPR) [Karpukhin et al., 2020], and it is widely adopted in the information retrieval domain (e.g., Lin et al. [2021], Moro and Valgimigli [2021]). We choose the DPR method because it does not interrupt the backpropagation, differently from other solutions, e.g., BM25, TF-IDF [Domeniconi et al., 2014b] or LSA [Domeniconi et al., 2016b].

We formalize this step as $B_\beta(\mathbf{C}_i) = E$, where B is a BERT-based model, β represents its parameters, and E is a matrix of shape $(len(\mathbf{C}_i), 768)$, where each row j of the matrix is the latent representation of the document c_j.

4.1.3.2 Discriminator

The main idea of the *Discriminator* is to discriminate the critical information from noise in a cluster of topic-related documents with respect to a shared background without breaking the backpropagation chain. For this reason, we use a probabilistic deep neural model to draw a probability distribution over documents in the cluster $< c_0, c_1, ..., c_n > \in \mathbf{C}_i$, with the following formula:

$$p_\theta(\mathbf{C}_i|x_i) \tag{4.2}$$

where θ represents the parameters of the neural network. Even in this case, the neural model is a BERT-based pre-trained language model as the one used for indexing, but this is trained during the learning process while the first is frozen. In detail, the *Discriminator* creates a latent projection for each background, which is used to fetch the more related documents in the cluster. More precisely, it applies the inner product to create a score for each document and selects the top K ones.

4.1.3.3 Generator

We use the pre-trained encoder-decoder generative transformer BART [Lewis et al., 2020a] to summarize the $\mathbf{C}_i$ weighted by the *Discriminator*. This component is trained to predict the next output token, creating a probability distribution over the dictionary for each $c_j \in \mathbf{C}_i$ before marginalizing. The process is then repeated for all the target tokens.

Before giving the documents to the model, we concatenate them with the background x_i, creating $c'_{ij} = [x_i, tok, c_{ij}]$, where *tok* is a special text separator token (*¡doc¿*) we add between x_i and c_{ij} to make BART aware of the background text boundary. The behavior of the *Generator* can be formally defined as follows:

$$p(y_i|c'_{ij}) = \prod_{z}^{N} p_\gamma(y_{iz}|c'_{ij}, y_{i,1:z-1}) \tag{4.3}$$

where γ are the *Generator* parameters, $N = |y_i|$ is the target length, and $y_{i,1:z}$ are the tokens from position 1 to z of the target y_i.

4.1.3.4 Model

The entire model aims to draw the probability distribution over the dictionary to generate the output tokens y_i conditioned by x_i and $\mathbf{C_i}$ that we formally define as:

$$p(y_i|x_i, \mathbf{C}_i) = \prod_{z}^{N} \sum_{c \in top-k} p_\theta(c_i|x_i) p_\gamma(y_{iz}|c'_i, y_{i,1:z-1}) \tag{4.4}$$

We train the whole model by minimizing the negative marginal log-likelihood of each target with the following loss:

$$-\sum \log p(y|x, \mathbf{C}) \tag{4.5}$$

4.1.4 Experiments

This section starts with describing the dataset in Section 4.1.4.1 and training details in Section 4.1.4.2. We then analyze model performance in Section 4.1.4.3 and finally conduct ablation studies in Section 4.1.4.4.

4.1.4.1 Dataset

We tested and evaluated our proposed method on the only medical dataset for multi-document summarization, as far as we know, about the generation of systematic literature reviews: the MS2 dataset. The dataset is provided in DeYoung et al. [2021], and it is freely distributed.[2] It contains over 470K document abstracts and 20K summaries derived from the scientific literature. Each sample of the dataset is composed of three elements: i) the *background* statement, which is a short text that describes the research

[2] https://github.com/allenai/ms2

TABLE 4.1. The dataset statistics. All values are mean over the whole dataset except for the "# backgrounds" row.

Statistic	Ms2 Background	Ms2 Document	Ms2 Summary
# average num. of tokens	125.75	546.90	74.52
# average num. of abstracts per background		23.30	
# backgrounds		13982	

question or topic shared by all documents in the cluster, ii) the *target* statement, which is the multi-document summary to generate, and iii) the *studies*, also defined as cluster for consistency with our notation, which is a set of abstracts of medical studies related to the topic covered in the background statement.

The problem can be formalized as follows: we have a target statement to generate about the background source, containing the topic specifications, and a cluster of related document abstracts from which to fetch and discriminate helpful knowledge with respect to the background. From here on, we use the terms "document" and "abstract" interchangeably since the elements in the cluster are just the abstracts of medical documents.

We report the dataset statistics in Table 4.1.

4.1.4.2 Training Details

We trained our solution for 3 epochs using a batch size of 1 and a learning rate with a linear schedule set to 1×10^{-5}. We set the number of K equal to 6 because it gave best results and used 1024 tokens as the max input size for the *Generator*. During the evaluation, we adopted a beam size of 4 with a min and a max length set to 32 and 256, respectively.

We implemented the code using PyTorch for tensor computations and Hugging Face[3] for language model checkpoints. We performed the experiments on a workstation with a GPU Nvidia RTX 3090 of 24GB memory, 64GB of RAM, and a processor Intel(R) Core(TM) i9-10900X CPU @ 3.70GHz.

4.1.4.3 Results

Table 4.2 shows the results on multi-document summarization of systematic literature reviews, comparing our method with two solutions proposed in DeYoung et al. [2021]. The $\text{BART}_{\text{HIERARCHICAL}}$ solution is trained to encode each document independently and then concatenate the representation of hidden states before decoding, whereas LED_{FLAT} takes as input all documents concatenated as a single document. Experimental results show we outperform the state-of-the-art in all the ROUGE metrics, proving better capability to discriminate relevant information across many related documents and merge it consistently (Figure 4.2).

[3] https://huggingface.co/models

TABLE 4.2. The results on MS2 for multi-document summarization of systematic literature reviews. The results of previous state-of-the-art are taken from DeYoung et al. [2021]. Better ROUGE scores are bolded.

	Ms2		
Model	Rouge-1	Rouge-2	Rouge-L
Prev. SOTA			
LED_{FLAT}	26.89	8.91	20.32
$\text{BART}_{\text{HIERARCHICAL}}$	27.56	9.40	20.80
Our			
DAMEN	**28.95**	**9.72**	**21.83**

4.1.4.4 Ablations

We conducted ablation studies on the MS2 dataset to prove the importance of each module of our method. In detail, for all experiments we trained our solution for 1 epoch with the same training details reported in Section 4.1.4.2, and we performed the evaluation on the first 400 instances of the test set.

The Importance of a Highly Abstractive Large-Sized Generator We report in Table 4.3 the performance using several pre-trained checkpoints of the *Generator* that differ in size and training. In detail, we tested two $\text{BART}_{\text{BASE}}$ checkpoints and three $\text{BART}_{\text{LARGE}}$ checkpoints:

- ***facebook/bart-base***: the actual BART model pre-trained with a denoising masked

Background: An individual patient data meta analysis was performed to determine clinical outcomes, and to propose a **risk stratification** system, related to the comprehensive treatment of patients with **oligometastatic nsclc**.

Doc1: ... **We therefore did this phase iii trial to compare concurrent chemotherapy and radiotherapy** followed by resection with st and ard concurrent chemotherapy and definitive radiotherapy without resection ... **In an exploratory analysis, os was improved for patients who underwent lobectomy, but not pneumonectomy, versus chemotherapy plus radiotherapy. Chemotherapy plus radiotherapy** with or without resection (preferably lobectomy) are options for patients with stage iiia (n2) non-small-cell lung cancer.

Doc2: ... Common **adverse events** associated with crizotinib were visual disorder, gastrointestinal side effects, and elevated liver aminotransferase levels, whereas common adverse events with chemotherapy were fatigue, alopecia, and dyspnea. **Patients reported greater reductions in symptoms of lung cancer and greater improvement in global quality of life with crizotinib than with chemotherapy**.

Doc3: ... First-line gefitinib for patients with advanced non-small-cell lung cancer who were selected on the basis of egfr mutations **improved progression-free survival**, with acceptable toxicity, as **compared with st and ard chemotherapy** ...

Ground-truth: Significant os differences were observed in oligometastatic patients stratified according to type of metastatic presentation, and n status. Long-term survival is common in selected patients with metachronous oligometastases.

Model: The pooled **risk stratification** of patients with **oligometastatic nsclc** showed a **significant reduction in the risk of adverse events compared with st and ard chemotherapy**, **but not radiotherapy**.

FIGURE 4.2. A random sampled test set instance. We show how DAMEN selects the information from the background and multiple documents to generate the final summary.

TABLE 4.3. The ablations to validate the contribution of the *Generator*. Better ROUGE scores are bolded.

	Ms2		
Generator	Rouge-1	Rouge-2	Rouge-L
$\text{Bart}_{\text{BASE}}$			
Original MLM	26.84	7.90	20.45
Scientific MLM	25.81	7.65	19.96
$\text{Bart}_{\text{LARGE}}$			
Original MLM	24.81	7.21	19.31
CNN	27.22	8.30	20.99
XSUM	**28.35**	**8.96**	**21.62**

language modeling.

- ***gayanin/bart-mlm-pubmed***: the BART model pre-trained exclusively on scientific corpora.
- ***facebook/bart-large***: the same BART model as the base version with a large architecture.
- ***facebook/bart-large-cnn***: the large BART fine-tuned on single-document summarization on the CNN/DailyMail dataset [Nallapati et al., 2016].
- ***facebook/bart-large-xsum***: the large BART fine-tuned on single-document summarization on the XSUM dataset [Narayan et al., 2018].

Results prove that a large-sized BART model already fine-tuned on a summarization task achieves better performance. More precisely, the checkpoint fine-tuned on the XSUM dataset obtains better results thanks to the higher abstractiveness and the shortness of the target summaries, which are made up of just 1-2 sentences, similar to the MS2 dataset.

The Importance of a Full-Sized Chunked Representation of Documents in the Cluster Table 4.4 reports experiments with three cluster configurations, where each document is treated with a different text representation, described as follows:

- ***Document-level***: the simpler configuration that considers the entire abstracts in the cluster. We truncated documents taking only the first 512 tokens before encoding by the *Indexer*.
- ***Sentence-level***: we considered the sentences of each document obtained using the state-of-the-art tokenizer PySBD [Sadvilkar and Neumann, 2020]. The sentences are encoded up to 128 tokens in length and they are then treated as individual textual units.

TABLE 4.4. The ablations to validate the best cluster configuration. Better ROUGE scores are bolded.

	Ms2		
Cluster	Rouge-1	Rouge-2	Rouge-L
Baselines			
Sentence-level	27.77	8.82	21.11
Document-level	28.14	8.82	21.33
Our			
Chunk-level	**28.35**	**8.96**	**21.62**

- ***Chunk-level***: our configuration, where each document is split into chunks of exact 512 tokens to consider all text information without input truncation. This configuration is similar to the "sentence-level" one but with the difference that each textual unit is 512 tokens in length and not 128.

The results prove the better performance on a cluster with chunked documents. By considering 512 tokens for each document, we fully leverage the capability of BERT language modeling without truncating any information. Input truncation required by the "document-level" configuration plays an important role in final accuracy because it discards and ignores potential summary-relevant information, leading to a performance drop. The "sentence-level" setting lets us increase the top K sentences to retrieve, but it worsens the final summary because single sentences are too fine-grained.

The Importance of a Background-First Concatenation with Special Token Table 4.5 reports the experiments with a different configuration of the concatenated inputs to give to the *Generator*. We experimented with four types of concatenation:

- *[Document + Background]*
- *[Background + Document]*
- *[Document + <doc> + Background]*
- *[Background + <doc> + Document]*

Results prove the importance of a background-first concatenation with the special token separator to make BART aware of the textual difference between the background and the documents.

The Importance of Pre-trained DPR Encoders. Table 4.6 reports the experiments with different model checkpoints for the *Indexer* and *Discriminator*. First, we leveraged the checkpoint "*sentence-transformers/allenai-specter*" [Cohan et al., 2020], which is a scientific BERT-based model trained to create document embeddings by using paper citations. Thus, we used this pre-trained model for both the *Indexer* and *Discriminator*. Second, we used two different checkpoints with a specific DPR training,

TABLE 4.5. The ablations to validate the input to give to the *Generator*. Better ROUGE scores are bolded.

	Ms2		
Concatenation	Rouge-1	Rouge-2	Rouge-L
w/o token <doc>			
Document + Background	28.01	8.83	21.18
Background + Document	27.70	8.63	20.98
w/ token <doc>			
Document + Background	27.96	8.65	21.43
Background + Document	**28.35**	**8.96**	**21.62**

such as "*facebook/dpr-question_encoder-single-nq-base*" for encoding the background and "*facebook/dpr-ctx_encoder-single-nq-base*" for encoding each document in the cluster.

Results prove the importance of the DPR checkpoints for both the *Indexer* and *Discriminator*.

4.1.5 Conclusion

We proposed a novel probabilistic method based on the combination of three language models to tackle multi-document summarization in the medical domain. This task is characterized by redundant information, noise, and the possible presence of vital information in each sentence that makes arbitrary input truncation unacceptable. For this reason, we proposed a multi-document summarization method able to discriminate salient contents from irrelevant before summarizing. In detail, the solution first leverages a BERT-based model (*Indexer*) for creating dense indices for each chunk of each document in the cluster. Then, a second BERT-based model (*Discriminator*) is used to process the shared background and select only the most relevant chunks. The final BART model is trained to perform a probability marginalization over each token prediction for each selected chunk. In this way, our solution reads all document information and selects just the most relevant chunks, discarding noise before feeding the *Generator*. The *Discriminator* and *Generator* are trained end-to-end, backpropagating the probability distribution as explained in Section 4.1.3. The *Indexer* is frozen; training would lead to

TABLE 4.6. The ablations to validate the contribution of the *Indexer* and *Discriminator* model checkpoints. Better ROUGE scores are bolded.

	Ms2		
Encoders	Rouge-1	Rouge-2	Rouge-L
w/o ad-hoc encoders			
SPECTER-based	27.79	8.60	21.23
w/ ad-hoc encoders			
DPR-based	**28.35**	**8.96**	**21.62**

some problems, such as the time to learn improved embeddings at each iteration and the larger memory occupation to save the gradient for each document.

We tested our method on Ms2, the only dataset on systematic literature reviews, and compared it with state-of-the-art models, finding that our novel approach outperforms competitors on the ROUGE evaluation metrics. Further, we performed extensive ablation studies to highlight the contribution of each component and motivate the design choices.

4.1.5.1 Future Works

At the edge of our knowledge, this is the first work that applies a probability marginalization method for multi-document summarization. We believe this work can inspire novel research towards end-to-end multi-model collaboration instead of solutions with a single large model addressing the entire task. According to the *divide et impera* pattern, each model learns a specific sub-task, creating a more efficient and transparent cooperating solution. Tasks such as related work generation or text generation from multi-sourced inputs can get the most from our method, improving pre-existing solutions to discriminate helpful knowledge from noise.

Further possible directions to deal with multi-inputs are the following: i) extracting relevant snippets from documents with term weighting techniques [Domeniconi et al., 2015b] or semantic relations with unsupervised methods [Domeniconi et al., 2016c, 2017] to better model interpretable representations based on knowledge graph learning techniques [Frisoni and Moro, 2020, Chen et al., 2021a,c] or event extraction methods [Frisoni et al., 2021]; ii) training models to write and read cross-document information with self-supervised representation learning methods [Domeniconi et al., 2014c] and memory-based neural layers [Moro et al., 2018, Cui and Hu, 2021].

4.1.6 Ethical Considerations

The advancement of deep neural network architectures and the availability of large pre-trained language models has led to significant improvements for the multi-document summarization task, which has applications in high-impact domains, particularly in the medical one. Here, systematic literature reviews play an essential role for the medical and scientific community, and for that reason, they require strong guarantees about the factuality of the output summary. Current state-of-the-art NLP solutions cannot establish such assurance, so we do not believe our solution, like previous ones, is ready to be deployed. The research should explore more effective evaluation measures for text summarization to make it happen, and large-scale accuracy guarantees by medical experts are still needed. Finally, if the method will be applied to sensitive data such as medical patient records, it should also include privacy-preserving policies [da Silva et al., 2006].

4.2
Retrieve-and-Rank End-to-End Summarization of Biomedical Studies

Originally published as: *Gianluca Moro, Luca Ragazzi, Lorenzo Valgimigli, and Lorenzo Molfetta. Retrieve-and-rank end-to-end summarization of biomedical studies. In Oscar Pedreira and Vladimir Estivill-Castro, editors, Similarity Search and Applications - 16th International Conference, SISAP 2023, A Coruña, Spain, October 9-11, 2023, Proceedings, volume 14289 of Lecture Notes in Computer Science, pages 64–78. Springer, 2023e. doi: 10.1007/978-3-031-46994-7_6. URL https://doi.org/10.1007/978-3-031-46994-7_6*

Abstract

An arduous biomedical task involves condensing evidence derived from multiple interrelated studies, given a context as input, to generate reviews or provide answers autonomously. We named this task context-aware multi-document summarization (CA-MDS). Existing state-of-the-art (SOTA) solutions require truncation of the input due to the high memory demands, resulting in the loss of meaningful content. To address this issue effectively, we propose a novel approach called Ramses, which employs a retrieve-and-rank technique for end-to-end summarization. The model acquires the ability to (i) index each document by modeling its semantic features, (ii) retrieve the most relevant ones, and (iii) generate a summary via token probability marginalization. To facilitate the evaluation, we introduce a new dataset, FAQsumC19, which includes the synthesizing of multiple supporting papers to answer questions related to Covid-19. Our experimental findings demonstrate that Ramses achieves notably superior ROUGE scores compared to state-of-the-art methodologies, including the establishment of a new SOTA for the generation of systematic literature reviews using Ms2. Quality observation through human evaluation indicates that our model produces more informative responses than previous leading approaches.

4.2.1 Introduction

Given the paramount societal role of biomedicine and related natural language processing (NLP) tasks [Frisoni et al., 2020a, Frisoni and Moro, 2020, Frisoni et al., 2021, Moro and Valgimigli, 2021, Frisoni et al., 2022c, 2023], aggregating information from multiple topic-related biomedical papers to help search, synthesize, and answer questions is of great interest [DeYoung et al., 2021]. Real-world applications require indexing, combining, and summarizing evidence from clinical trials on a research background to produce systematic literature reviews (SLRs) or answer medical inquiries. Consequently, we define such activities as *context-aware multi-document summarization* (CA-MDS) due to

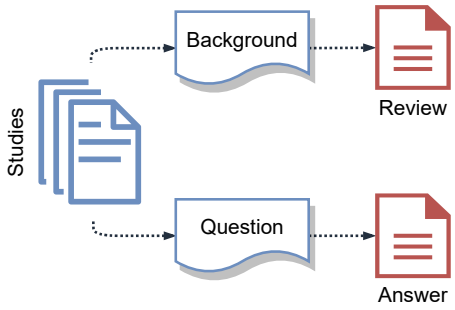


FIGURE 4.3. The overview of biomedical CA-MDS. In our experiments, we use the input contexts (i.e., the background or question) to retrieve the salient studies and aggregate them to generate the target (Input → Output). For example, given a set of topic-related scientific papers, the goal is to select the ones more correlated to a user's input question to produce a single answer.

the presence of an input context (i.e., background or question) that conditions the downstream summarization task (Figure 4.3). In real life, biomedical articles usually contain several thousands of words that compose lingo and complicated expressions, making understanding them a time- and labor-consuming process even for professionals. Thus, automation support for biomedical activities is practical and beneficial in facilitating knowledge acquisition.

CA-MDS solutions for biomedical applications should process all inputs without ignoring any details, reducing the risk of model hallucination, namely generating unfaithful outputs due to training on targets having facts unfounded by the source. Therefore, state-of-the-art (SOTA) models rely on sparse transformers [Beltagy et al., 2020], Fusion-in-Decoder strategies [Izacard and Grave, 2021], and marginalization-based decoding [Moro et al., 2022a]. However, such methods either (i) need high memory requirements that force input truncation for organizations operating in low-resource regimes [Moro and Ragazzi, 2022, Moro et al., 2023d, Moro and Ragazzi, 2023, Moro et al., 2023c,a], or (ii) lack end-to-end learning, reducing the potential of cooperating neural modules.

In this paper, we introduce RAMSES,[4] a retrieve-and-rank summarization approach trained via end-to-end learning to retrieve salient biomedical documents by their semantic meaning and synthesize them given an input context. RAMSES comprises a biomedical bi-encoder and a generative aggregator. The bi-encoder reads all the documents, represents their semantics via embeddings, and retrieves and scores salient documents related to an input context. Then, the aggregator is conditioned by the context along with these latent documents to decode the summary by marginalizing the token probability distribution weighted by their relevance score.

We evaluated RAMSES in two biomedical CA-MDS tasks: (i) producing SLRs on the MS2 dataset [DeYoung et al., 2021] and (ii) answering frequently asked questions (FAQs) about Covid-19 in our proposed dataset FAQSUMC19. In detail, we collected 514 Covid-19 FAQs with high-quality abstractive answers written by experts. Then we

[4] https://disi-unibo-nlp.github.io/projects/ramses/

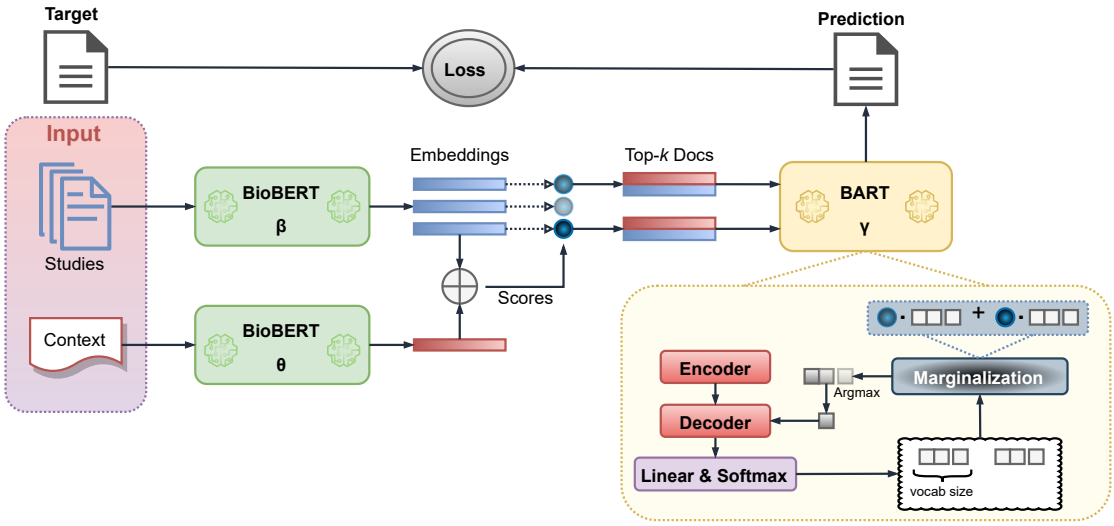


FIGURE 4.4. The overview of RAMSES. The input is a biomedical context and multiple studies that are encoded by two different BIOBERT models. Then, we compute the relevance score of each document conditioned by the context. Finally, the top-k most salient documents are concatenated with the context and given to BART that marginalizes their token probability distribution, weighted by the relevance scores, at decoding time.

augmented each instance with 30 supporting scientific papers containing the information needed to answer the question, producing 15,420 articles. In particular, FAQSUMC19 has two essential features: (i) includes abstractive answers authored by experts, unlike other related datasets that use extractive targets [Rajpurkar et al., 2018]; (ii) is the first CA-MDS dataset for Covid-19, becoming a crucial benchmark for producing multi-document summaries to answer questions on Covid-19 with the support of updated related biomedical papers.

We perform extensive experiments, showing that RAMSES achieves new SOTA performance in the MS2 dataset and outperforms previous solutions in FAQSUMC19, whose inferred answers are also rated as of more quality by human experts.

4.2.2 Related Work

Semantic Neural Retriever Applications The semantic representation skill exhibited by neural networks has catalyzed the emergence of groundbreaking neural methodologies in information retrieval [Formal et al., 2022] First, the algorithm BM25 has been exceeded by dense passage retrieval (DPR) [Karpukhin et al., 2020], a remarkable neural application that has since evolved into a fundamental element within numerous neural-driven retrieval solutions [Izacard and Grave, 2021, Ren et al., 2021b,a]. These neural retrievers have been fused with a language model to enrich and improve input [Lewis et al., 2020b], generating superior models characterized by increased efficiency and improved performance [Borgeaud et al., 2022, Frisoni et al., 2022c]. Despite their promising

results, the end-to-end application of these solutions in MDS remains unexplored.

NLP for Biomedical Documents Much recent work in NLP has concentrated on the biomedical domain [Moro and Masseroli, 2021], including CA-MDS [DeYoung et al., 2021], which can decrease the burden on medical workers by highlighting and aggregating key points while reducing the amount of information to read. Previous contributions focused on the automatic generation of SLRs. In detail, cutting-edge solutions rely on three different neural architectures: (i) transformer-based models with linear complexity in the input size thanks to sparse attention [Beltagy et al., 2020], which concatenate the input context along with all documents in the cluster producing a single source sequence; (ii) quadratic transformers with Fusion-in-Decoder [Izacard and Grave, 2021], which join the hidden states of documents after encoding them individually; (iii) marginalization-based decoding augmented by frozen retrievers [Moro et al., 2022a], which first pinpoints salient documents w.r.t. a query and produces a single summary by summing the probability distribution of the inferred token for each document.

MDS Solutions in other Domains Flat approaches with MDS-specific pre-training [Xiao et al., 2022] concatenate the sources in a single text, treating MDS as a single-input task. Hierarchical approaches merge document relations to obtain semantically rich representations by leveraging graph-based methods [Amplayo and Lapata, 2021] and multi-head pooling and interparagraph attention [Jin et al., 2020]. Marginalization-based approaches [Hokamp et al., 2020] apply marginalization to the token probability distribution at the decoding time to produce a single output from many inputs. The two-stage approaches [Liu and Lapata, 2019a] adopt different strategies to rank sources before producing the summary. Unlike previous work, RAMSES is trained in end-to-end learning to retrieve relevant text from biomedical articles and marginalize the probability distribution of the latent extracted information at decoding time.

Covid-19 Datasets With the appearance of Covid-19, thousands of articles have been published quickly. To aid experts in accessing this knowledge, large organizations collected corpora such as CORD-19 [Wang et al., 2020b] and LITCOVID [Chen et al., 2021b], encouraging the proposal of task-specific datasets. COVID-QA [Möller et al., 2020] study question-answering using annotated pairs extracted from 147 papers. COVID-Q [Wei et al., 2020] collects 16,690 questions about Covid-19, classifying them into 15 categories. Poliak et al. [2020] scrapped over 40 trusted websites for Covid-19 FAQs, creating a collection of 2100 questions. Zhang et al. [2021] proposed a dataset for the retrieval of FAQs, where user queries are semantically paired with existing FAQs. FAQSUMC19 fills this gap, introducing the first CA-MDS dataset to answer Covid-19 FAQs by summarizing multiple related studies.

Fine-grained comparisons with previous work are in Section 4.2.6.1.

4.2.3 Preliminary

We provide details for context-aware multi-document summarization (CA-MDS).

Definition CA-MDS aims to compile a summary from a cluster of related articles given an input context, analogous to the query in query-focused summarization [Vig et al., 2022]. Yet, unlike answering FAQs, SLR generation does not consider questions. Thus, we define the task we face as CA-MDS. The biomedical tasks we address in this work, such as SLR generation and FAQ answering, are CA-MDS tasks because they both have an input context (i.e., the research issue in SLRs and the human question in FAQs) and many topic-related documents from which produce the output.

Problem Formulation In the CA-MDS setting, we have $(c, \mathbf{D}, y)$, where c is the input context, $\mathbf{D}$ is the cluster of topic-related documents, and y is the target generated from $\mathbf{D}$ given c. Formally, we want to predict y from $\{c, d_1, ..., d_n | d \in \mathbf{D}\}$.

4.2.4 Method

The end-to-end learning of RAMSES allows the cooperating modules to jointly retrieve and aggregate key information from multiple sources in one output (Figure 4.4).

Given the context c and the documents $\mathbf{D}$, our method first generates relevance scores on $\mathbf{D}$ with a biomedical solution based on DPR Karpukhin et al. [2020]:

$$p_{\beta,\theta}(d \in \mathbf{D}|c) = (Enc_\beta(d) \oplus Enc_\theta(c)) \tag{4.6}$$

where Enc_β and Enc_θ are two different BIOBERT-base models trained to produce a dense representation of documents and the context [Papanikolaou et al., 2022], respectively, $\oplus$ is the inner product between them, and $p(d|c)$ is the relevance score associated to the document d given c. Thus, our solution finds the most top-k relevant texts according to c. Then, given c and each $d \in$ top-k, a BART-base model [Lewis et al., 2020a] draws a distribution for each next output token for each d, before marginalizing:

$$p(y|c, \mathbf{D}) = \prod_{z}^{N} \sum_{d \in \text{top-}k} p_\theta(d|c) p_\gamma(y_z | d^{'}, y_{1:z-1}) \tag{4.7}$$

where $d^{'} = [c, tok, d]$ is the concatenation of c and $d \in$ top-k with a special text separator token (<doc-sep>) to make the model aware of the textual boundary, N is the target length, and $p_\gamma(y_z | d^{'}, y_{1:z-1})$ is the probability of generating the target token y_z given d' and the previously generated tokens $y_{1:z-1}$.

We train our RAMSES model by minimizing the negative marginal log-likelihood of each target with the following loss function:

$$\mathcal{L} = -\sum_{i} \log p(y_i | c_i, \mathbf{D}_i) \tag{4.8}$$

End-to-End Learning The model (Equation 4.7) allows the gradient to backpropagate to all modules. For clarity, we rewrite the formula as a continuous function, as follows:

$$\text{RAMSES}(\mathbf{D}, c) = \sum_{(d_j, s_j)} B_\gamma([c, tok, d_j]) \cdot s_j \tag{4.9}$$

$$\text{top-}k(\mathbf{D}, c) = [(d_1, s_1), \ldots, (d_k, s_k)] \tag{4.10}$$

$$s_j = Enc_\beta(d_j) \oplus Enc_\theta(c) \tag{4.11}$$

where $(d_j, s_j) \in$ top-k and B_γ is BART.

The presence of s_j in Equation 4.9 allows the gradient, computed by minimizing the objective function, to reach Enc_β and Enc_θ. For this reason, the documents and context embeddings are adjusted during the training to improve the generated summary, making all modules of our solution learn jointly in an end-to-end fashion.

4.2.5 FAQsumC19 Dataset

We introduce a new dataset, FAQSUMC19, containing 514 Covid-19-related FAQs with abstractive answers written by experts, each supported by 30 abstracts of scientific articles, for a total of 15,420 documents. We obtained from the Covid-19 FAQ section on WHO[5] all available question-answer pairs. We then augmented each instance with 30 Covid-19 scientific articles strictly related to the question from the updated version of the CORD-19 dataset [Wang et al., 2020b]. Specifically, we experimented with the selection of supporting articles with different information retrieval methods, such as a random baseline, BM25 [Robertson and Walker, 1994], and SUBLIMER [Moro and Valgimigli, 2021]. We used the concatenation between the question and the answer to retrieve the first 30 ranked documents regarding semantic similarity, creating a knowledge base to support the answer generation. We finally split the dataset into 464 instances for training ($\approx 90\%$) and 50 for the test ($\approx 10\%$).

To assess the quality of question-cluster pairs in our dataset, we computed the content coverage with ROUGE-1 precision [Lin, 2004] and BERTScore [Zhang et al., 2020b] of the question-answer concatenation w.r.t. each document in the cluster, and calculate the average score. We evaluate the syntactic and semantic overlap between the question and answer and the texts. Table 4.7 reveals that SUBLIMER achieves the best scores, as expected.

4.2.6 Experiments

4.2.6.1 Experimental Setup

Datasets Table 4.8 reports the statistics of the datasets used to test RAMSES in different biomedical tasks: **Ms2** [DeYoung et al., 2021] consists of 15,597 instances derived

[5] https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub

TABLE 4.7. The question-cluster pairs' quality. Best values are bolded.

	ROUGE			BERTScore		
	Avg	Max	Min	Avg	Max	Min
RANDOM	12.34	23.21	0.17	11.30	24.71	2.80
BM25	16.70	29.68	0.37	16.17	35.20	1.13
SUBLIMER	**20.44**	**33.31**	**2.32**	**21.11**	**36.79**	**5.75**

from the scientific literature. Each sample is composed of (i) the background statement, which describes the context research issue, (ii) the target statement, which is the summary to generate; and (iii) the studies, which are the abstracts of biomedical documents that contain the needed information for the research issue. **FAQsumC19** is our proposed dataset that comprises 514 Covid-19 FAQs with abstractive answers written by experts, each supported by 30 abstracts of scientific papers.

Baselines We compare RAMSES with SOTA solutions: **Bart-FiD** [DeYoung et al., 2021], which is BART with the Fusion-in-Decoder strategy [Izacard and Grave, 2021], encodes all sources individually and combines their hidden states before decoding. **Led-Gaq** [DeYoung et al., 2021], which is LED [Beltagy et al., 2020] with global attention on the input query, concatenates all texts in a single input of up to 16,384 tokens. **Damen** [Moro et al., 2022a], a retrieval-enhanced solution with marginalization-based decoding, discriminates important fragments of the cluster with a frozen BERT-base model and marginalizes their probability distribution during decoding. **Primera** [Xiao et al., 2022], which is LED pre-trained with a multi-document summarization-specific objective, concatenates the texts with a special separator token up to 4096 tokens in size.

Evaluation Metrics We use ROUGE-1/2/L [Lin, 2004] to assess fluency and informativeness. We also adopt $\mathcal{R}$ [Moro et al., 2023b] as an aggregated judgment that considers the variance of the ROUGE scores. Finally, we perform qualitative analysis to bridge the superficiality of automatic evaluation measures.

TABLE 4.8. The datasets used for evaluation (FAQSUMC19 is ours). Statistics include dataset size and the average (i) number of source (S) documents per instance, (ii) number of total words in S and target (T) texts, and (iii) S-T compression ratio of words [Grusky et al., 2018].

		S		T	S → T
Dataset	**Samples**	Docs	Words	Words	Comp
MS2	15,597	23.30	9563.95	70.81	135.06
FAQSUMC19	514	30	5635.50	139.06	40.53

TABLE 4.9. Performance of models on the evaluation datasets. The best scores are in bold.

	Ms2				FAQsumC19			
Model	**R-1**$_{f1}$	**R-2**$_{f1}$	**R-L**$_{f1}$	$\mathcal{R}$	**R-1**$_{f1}$	**R-2**$_{f1}$	**R-L**$_{f1}$	$\mathcal{R}$
Baselines								
LED-GAQ	26.89	8.91	20.32	18.60	25.55	4.42	13.77	14.47
BART-FID	27.56	9.49	20.80	19.18	20.26	5.59	14.84	13.51
DAMEN	28.95	9.72	21.83	20.04	23.81	3.50	13.03	13.35
PRIMERA	30.07	9.85	22.16	20.55	25.04	3.64	13.00	13.79
Our								
RAMSES	**31.83**	**10.44**	**22.19**	**21.32**	**30.18**	**7.31**	**15.67**	**17.56**

Implementation We fine-tune the models using PyTorch and the HuggingFace library, setting the seed to 42 for reproducibility. RAMSES is trained on an NVIDIA RTX 3090 GPU of 24 GB memory from an internal cluster for 1 epoch with a learning rate of 3e-5 on MS2 and for 3 epochs with a learning rate of 1e-5 on FAQSUMC19. For decoding, we use the beam search with 4 beams and the following min-max target size: 32-256 for MS2 and 100-256 for FAQSUMC19.

4.2.6.2 Results

Table 4.9 reports the performance of the models in the two evaluation datasets. RAMSES yields better scores, suggesting that the retrieve-and-rank end-to-end learning is more effective than prior SOTA approaches in both biomedical CA-MDS tasks.

The Impact of k As our method relies on learning to select the best top-k relevant documents from the cluster, the value of k is crucial for model performance and GPU memory occupation. Therefore, we analyze the impact of k on model performance by experimenting with a different number of documents to retrieve: 3, 6, 9, 12, 15, 18. Table 4.10 reports a slight performance improvement as k increases until a threshold is reached (e.g., $k = 9$ for BART-base), indicating that the marginalization approach with

TABLE 4.10. ROUGE F1 scores (R-1, R-2, R-L) on MS2 on evaluating RAMSES with different generator checkpoints (B and L stand for base and large, respectively) and k documents retrieved at training time. OOM means "GPU out of memory exception." The best results are bolded.

	Ms2		
k	BART-B	BART-L	PEGASUS-L
3	31.14/9.78/21.43	31.93/10.40/21.96	27.83/7.27/18.93
6	31.12/9.88/21.68	**31.97/10.58/22.15**	28.71/8.26/19.39
9	31.81/10.30/22.13	31.00/10.17/21.76	28.61/8.32/19.35
12	31.15/10.14/21.58	31.10/10.09/6.52	27.58/7.30/18.51
15	30.94/9.82/21.39	31.72/10.53/21.90	OOM
18	31.36/10.20/21.75	31.81/10.43/21.87	OOM

TABLE 4.11. The results of RAMSES on MS2 by varying k at inference time and on FAQSUMC19 by varying k at training time. The best scores are bolded.

	Ms2			**FAQsumC19**		
k	**R-1**$_{f1}$	**R-2**$_{f1}$	**R-L**$_{f1}$	**R-1**$_{f1}$	**R-2**$_{f1}$	**R-L**$_{f1}$
3	30.98	9.75	21.48	29.32	7.05	15.71
6	31.49	10.14	21.94	28.69	6.51	15.20
9	31.81	10.30	22.13	28.74	6.83	15.44
12	**31.83**	**10.44**	**22.19**	**30.18**	**7.31**	**15.67**
15	31.73	10.36	22.10	29.14	6.65	15.26
18	31.72	10.39	22.04	29.06	6.89	15.31

more documents helps produce better ROUGE scores. However, a high k (i.e., $k \geq 12$) can also increase information redundancy and contradiction, lowering the final performance. Table 4.10 also lists the results of different models on single text summarization as the aggregator's checkpoint, such as BART and PEGASUS [Zhang et al., 2020a]. We notice that BART-large achieves better ROUGE scores, although PEGASUS is the largest model. However, as BART-base achieved a slightly lower result despite the noticeably fewer trainable parameters, we chose to use it for all experiments. Therefore, we tested the best checkpoint of BART-base trained with $k = 9$ with a different k at the inference time in MS2. Table 4.11 reports that the best performance has been achieved with $k = 12$. Furthermore, Table 4.11 also shows the results on FAQSUMC19 with a different k at training time, revealing a trend similar to MS2.

Memory Requirements Figure 4.5 shows the memory complexity at the training time of RAMSES for each k. We notice that the memory occupation is linear w.r.t. k, indicating that our solution is not computationally expensive, even for large clusters.

4.2.6.3 Ablation Studies

Table 4.12 reports the ablation studies on MS2 using RAMSES with BART-base and $k = 9$ with the same hyperparameter settings for all experiments.

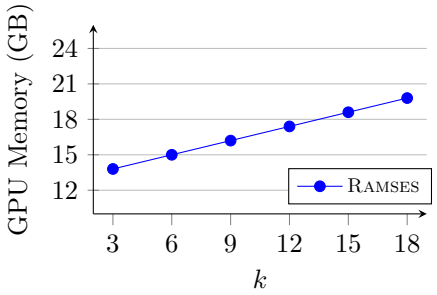


FIGURE 4.5. The RAMSES's GPU memory requirements by varying k at training time.

TABLE 4.12. The ablation studies on MS2. We gradually remove each module of RAMSES to show the performance drop. The best scores are in bold.

	Ms2			
	R-1$_{f1}$	**R-2**$_{f1}$	**R-L**$_{f1}$	$\mathcal{R}$
RAMSES	**31.83**	**10.44**	**22.19**	**21.32**
w/o context-first	31.57	10.26	21.89	21.08
w/o trained-retrieval	31.24	10.03	21.77	20.86
w/o inner-product	31.12	10.10	21.82	20.86
w/o token-sep	31.03	10.12	21.77	20.82
w/o bi-encoder	31.47	9.88	21.52	20.79
w/o context	25.26	5.33	17.37	15.88

Excluding the input context from the input concatenation to give to the generative aggregator (***w/o* context**) leads to the most significant decrease in performance. Indeed, the context is the research question shared by all documents in the cluster, so it contains important information for the final summary.

Training a single model to encode both the context and documents (***w/o* bi-encoder**), namely using a shared BIOBERT model, decreases performance. Indeed, since the context and the documents have two different purposes (i.e., we need the context to select context-related documents), two models are needed to specialize and differentiate the text representation. Despite the similarity of the two tasks, they differ for two main reasons: (i) the context is relatively shorter than the documents in the cluster, and (ii) the concept expressiveness is denser in the context than in the more verbose documents.

Removing the token separator <`doc-sep`> between the context and document (***w/o* token-sep**) decreases performance. Indeed, this token is needed to make the model aware of the textual boundary between the context and the documents.

Using cosine similarity instead of the inner product (***w/o* inner-product**) to score the documents against the input context achieves the worst results.

Freezing Enc_{β} (***w/o* trained-retrieval**) decreases performance, highlighting the usefulness of end-to-end learning to allow the model to select more informative documents.

Switching the context with the documents in the input concatenation (***w/o* context-first**) decreases performance, indicating that the leading context in the input helps the generative aggregator to focus better on how to join context-related information.

4.2.6.4 Human Evaluation

Considering the drawbacks of automatic metrics such as ROUGE [Fabbri et al., 2021], which is still the standard for evaluating text generation, we qualitatively evaluated the answers inferred from the FAQs of the entire FAQSUMC19 test set with three domain experts with master's degrees in medical and biological areas.

TABLE 4.13. The human evaluation on FAQSUMC19 with inter-annotator agreement (IAA) using WHO ground-truth answers and the inferred ones by RAMSES and LED-GAQ. `A>B` means how many times the generated answer from `A` was scored higher than `B`.

	Human Evaluation		
	RAMSES>LED	RAMSES>WHO	LED>WHO
Eval 1	84%	0%	0%
Eval 2	72%	20%	14%
Eval 3	72%	2%	0%
AVG	76%	7.3%	4.7%
IAA	46%	80%	86%

Instructions We gave evaluators a table, with each row containing the question and three possible answers in random order: (i) the "gold" from WHO, (ii) the prediction of RAMSES, and (iii) the prediction of LED-GAQ (the second-best model in FAQSUMC19 according to $\mathcal{R}$). Each expert was asked to order the answers according to how thoroughly they answered the question, focusing primarily on the factuality. For fairness, we did not inform the evaluators about the answers' origins and the test's goal. Overall, experts completed the task in two days, reporting no difficulty ordering the 50 answers.

Results Evaluation results, reported in Table 4.13, show that our method produces better informative abstractive answers to a given open question than a linear transformer with sparse attention. To be precise, the experts rated 76% of the answers of our solution as better than those of LED, with 46% agreement between the annotators (which means that 46% of the time, the three evaluators agree). Furthermore, the evaluators also found that 7.33% of the answers inferred by RAMSES are more informative than "gold" from WHO. Nevertheless, model generations are still far from being as informative as gold answers, indicating the limitations of current neural language models in FAQSUMC19.

4.2.7 Conclusion

In this paper, we introduced RAMSES, a retrieve-and-rank end-to-end learning solution for CA-MDS of biomedical studies. RAMSES is designed to simultaneously acquire indexing capabilities and retrieve pertinent documents to generate comprehensive summaries. Through multiple experiments on two biomedical datasets (including our proposed FAQSUMC19 to answer Covid-19 FAQs), we found that RAMSES outperforms SOTA models. This finding suggests that the integrated retrieval mechanism significantly benefits the CA-MDS task. Yet, human assessments indicate that there is still notable room for improvement, motivating further research in pursuit of novel retrieval applications within the realm of biomedical multi-document summarization.

Future works can investigate and include multimodal [Moro and Salvatori, 2022, Moro et al., 2023f], cross-domain [Domeniconi et al., 2017], and knowledge propagation [Lodi et al., 2010, Cerroni et al., 2013, 2015] approaches.

Chapter 5

Knowledge Injection and Distillation

5.1
Graph-based Abstractive Summarization of Extracted Essential Knowledge for Low-Resource Scenarios

Originally published as: *Gianluca Moro, Luca Ragazzi, and Lorenzo Valgimigli. Graph-based abstractive summarization of extracted essential knowledge for low-resource scenarios. In Kobi Gal, Ann Nowé, Grzegorz J. Nalepa, Roy Fairstein, and Roxana Radulescu, editors, ECAI 2023 - 26th European Conference on Artificial Intelligence, September 30 - October 4, 2023, Kraków, Poland - Including 12th Conference on Prestigious Applications of Intelligent Systems (PAIS 2023), volume 372 of Frontiers in Artificial Intelligence and Applications, pages 1747–1754. IOS Press, 2023c. doi: 10.3233/FAIA230460. URL https://doi.org/10.3233/FAIA230460*

Abstract

Although current summarization models can process increasingly long text sequences, they still struggle to capture salient related information spread across the lengthy size of inputs with few labeled training instances. Today's research still relies on standard input truncation without considering graph-based modeling of multiple semantic units to summarize only crucial facets. This paper proposes G-Seek, a graph-based summarization of extracted essential knowledge. By representing the long source with a heterogeneous graph, our method extracts and provides salient sentences to an abstractive summarization model to generate the summary. Experimental results in low-resource scenarios, distinguished by data scarcity, reveal that G-Seek consistently improves both the long- and multi-document summarization performance and accuracy across several datasets.

5.1.1 Introduction

The growing availability of unstructured information promotes text summarization solutions, which aim to generate concise synopses that convey the exact semantics of the source [Yadav et al., 2022]. Recently, due to the latest advancements in natural language processing (NLP)—from information retrieval [Moro and Valgimigli, 2021, Frisoni et al., 2022c, Moro and Salvatori, 2022, Moro et al., 2023f] to entity relationships acquisition [Frisoni and Moro, 2020, Frisoni et al., 2020a], classification [Domeniconi et al., 2017], and event extraction [Frisoni et al., 2021]—abstractive summarization [Beltagy et al., 2020, Lewis et al., 2020a, Frisoni et al., 2023] is receiving more interest, aiming to paraphrase the most meaningful details of documents instead of just retrieving them (i.e., extractive summarization). In this context, a particularly tough challenge is to process massive sequences, namely the task of long-input summarization. Two subtasks are long and multi-document summarization (LDS and MDS, respectively). LDS [Koh et al., 2023a] wants to capture and condense salient points scattered across a lengthy input (e.g., 10K words). In MDS [Ma et al., 2023], the synthesis is generated from a pool of sources related to the topic whose concatenation assembles a single long input to enable the summarization task to be addressed as in LDS [Xiao et al., 2022]. Thus, from now on, we refer to "the long input" as the source of both LDS and MDS tasks.

State-of-the-art (SOTA) solutions for long-input summarization are built upon transformers [Vaswani et al., 2017], which proficiently capture long-distance relations between words[1] with the self-attention mechanism. Nevertheless, such models are denoted by a structural constraint that proportionally links their memory usage to the input size, making them unduly resource-demanding when processing long texts. Consequently, this problem poses complications in LDS and MDS due to the long and intricate sources, which is even more noticeable in real-world low-resource scenarios [Moro and Ragazzi, 2023, Moro et al., 2023d] distinguished by the lack of labeled instances to supervise model training [Yu et al., 2021]. In fact, in realistic small and medium organizations, producing the gold summary of lengthy documents is costly, time-consuming, and may require domain knowledge specialists. For this reason, low-resource summarization is an important research topic that deserves more attention from the NLP community [Hedderich et al., 2021].

A promising approach to mitigate these obstacles is to use a semantic graph to represent the long input. Intuitively, by aggregating all source information, summary-worthy sentences (i.e., essential knowledge) can be pinpointed and extracted, avoiding standard input truncation and giving models more high-quality training samples that enable them to learn faster in data scarcity conditions. Previous contributions used graph representations for text summarization [Salchner and Jatowt, 2022]. However, their methods are proposed for extractive summarization [Jia et al., 2020a, Wang et al., 2020a, Chen et al., 2021a] and short texts Ji and Zhao [2021], Qiu and Cohen [2022] or do not take advantage of SOTA generative pre-trained language models (PLMs) [Wu et al., 2021b].

[1] Technically, these are subwords yielded by a subword tokenizer [Kudo and Richardson, 2018].

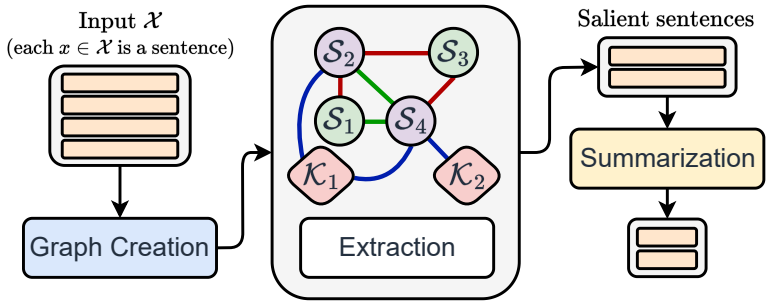


FIGURE 5.1. Overview of our approach. A long input $\mathcal{X}$ (each $x \in \mathcal{X}$ is a sentence) is converted into a heterogeneous graph: the pink diamonds represent the keyword nodes $\mathcal{K}$, the violet circles denote the sentence nodes $\mathcal{S}$ containing keywords, the green circles indicate the context of $\mathcal{S}$, and the different segments between nodes symbolize edges. The salient sentences are extracted and given to a generative PLM to produce the summary.

To fill this gap, we introduce G-SEEK (Figure 5.1),[2] a graph-based summarization of extracted essential knowledge. By representing the long source with a heterogeneous graph, G-SEEK extracts and gives the more relevant sentences to an abstractive summarizer for synthesis generation, avoiding feeding models only until their maximum input size, which prevents them from reading the overpart of the source, causing quality degradation [Moro and Ragazzi, 2022]. Technically, our graph employs a lightweight PLM to embody different semantic units (i.e., sentences and keywords) whose relationships are handled through multiple informative edges; then, a graph attention network (GAT) [Velickovic et al., 2018] is trained to select the salient nodes, soft labeled with a heuristic. Our solution is proposed explicitly for low-resource environments with a few dozen labeled training instances for the following motivations: (i) we show that giving more highly correlated source-target samples helps PLMs generate better summaries in a data scarcity scenario; (ii) the small number of trainable parameters of our solution (4M) lets G-SEEK not overfit over a few examples.

Extensive experimentations with multiple public datasets reveal that G-SEEK enhances the performance of SOTA summarization models in low-resource settings. Our contributions are as follows:

- We propose G-SEEK, a new low-resource long-input summarization approach that pinpoints and gives key sentences to generative PLMs by modeling the lengthy source with a semantic graph.
- We introduce a heterogeneous graph that captures the diversity of semantic units and their relationships through multiple edges.
- Experimental results in LDS and MDS datasets show that SOTA abstractive summarization models with G-SEEK improve syntactic and semantic evaluation metrics in a data scarcity scenario.

[2] https://disi-unibo-nlp.github.io/projects/gseek/

5.1.2 Related Work

Generative Pretrained Language Models Recent generative PLMs have shown strong performance and adaptation to LDS and MDS tasks. These models are based on the transformer encoder-decoder architecture [Vaswani et al., 2017] characterized by layers of self-attention. To be precise, quadratic PLMs such as BART [Lewis et al., 2020a] and PEGASUS [Zhang et al., 2020a] can process up to 1024 tokens due to the quadratic complexity w.r.t. the input size. Differently, linear PLMs such as LED [Beltagy et al., 2020] and PRIMERA [Xiao et al., 2022] can read more input (up to 4096 tokens with 24 GB of GPU memory), making them more suitable for long-input summarization. Although their impressive performance, such linear transformers, like quadratic ones, still rely on input truncation, namely processing the source until the model's maximum input size, ignoring potentially relevant summary-worthy details.

Graph-based Summarization Graphs and graph neural networks (GNNs) have played a thrilling role in MDS [Salchner and Jatowt, 2022], adding scalability [Jia et al., 2020b] and better domain modeling [Hogan et al., 2021] to mitigate the flaws of transformer-based models. Several contributions employ GNNs as standalone solutions [Muratore et al., 2010], where the summary is the composition of sentences extracted from the input [Doan et al., 2022]. Recent solutions are HETERSUMGRAPH [Wang et al., 2020a], which leverages a multilevel node representation, HAHSUM [Jia et al., 2020a], which exploits NER entities, and SGSUM [Chen et al., 2021a], which extracts subgraphs to generate the summary. On the contrary, GNNs can be embedded with abstractive summarization models to improve performance [Li et al., 2020]. BASS [Wu et al., 2021b] introduces a unified semantic graph to represent the group of texts and modifies the transformer architecture to interact with the graph. SKGSUM [Ji and Zhao, 2021] exploits nodes of different levels to guide the summary generation. Finally, HGSUM [Li et al., 2023] extends a linear transformer by incorporating a heterogeneous graph, training them jointly at the expense of a costly pipeline.

Other Approaches Two-stage methods [Gehrmann et al., 2018, Li et al., 2018, Liu et al., 2018, Liu and Lapata, 2019a] utilize various strategies to score and rank documents before generating the summary. Hierarchical solutions combine document relations to obtain semantic rich representations by leveraging graph-based techniques [Liao et al., 2018, Nayeem et al., 2018, Antognini and Faltings, 2019, Li et al., 2020, Amplayo and Lapata, 2021], multihead grouping and interparagraph attention [Jin et al., 2020, Liu and Lapata, 2019a], and maximal marginal relevance [Fabbri et al., 2019]. Marginalization-based techniques [Gu et al., 2020, Hokamp et al., 2020, Moro et al., 2022a, 2023e] apply marginalization during decoding to produce a single output from many inputs.

Although the effectiveness of all these different methods is similar, generative PLMs are still the SOTA approaches for long-input summarization due to (i) fast domain adaptation with few training data and (ii) more scalability and efficiency in computational resources.

Algorithm 6 Soft Labeling

Input:
$\mathcal{X} = \{x_1, \ldots, x_x\}$ ▷ Input sentences
$\mathcal{Y} = \{y_1, \ldots, y_y\}$ ▷ Output sentences
Parameters: $\mathcal{M}$ ▷ Similarity metric
Output: $\mathcal{S}$ ▷ Set of scores

$\mathcal{S} \leftarrow \emptyset$
for $x_i \in \mathcal{X}$ **do**
  $s \leftarrow \emptyset$
  for $y_i \in \mathcal{Y}$ **do**
    $s.append(\mathcal{M}(x_i, y_i))$
  end for
  $\mathcal{S}.append(max(s))$
end for
return $\mathcal{S}$

5.1.3 Method

In this section, we describe our novel *graph-based summarization of extracted essential knowledge* (G-SEEK) in detail. In a nutshell, G-SEEK comprises a document-to-graph module (blended with a GAT) and a pre-trained abstractive summarization model to generate a summary by focusing only on the essential input sentences. Section 5.1.3.1 explains the necessity of labeling the relevance of input sentences (i.e., soft labeling). Section 5.1.3.2 details the passages needed to construct the graph from the long input using semantic and structural information. Section 5.1.3.3 illustrates the GAT module devised to learn the relationships between graph nodes. Finally, Section 5.1.3.4 sums up the overall summarization pipeline of our proposed approach.

5.1.3.1 Soft Labeling

Recent advances in NLP have highlighted the need to identify summary-worthy sentences in the source to address the problem of processing large amounts of information [Bajaj et al., 2021, Mao et al., 2022]. Intuitively, each sentence can be labeled as relevant or not

TABLE 5.1. The MDS results on MULTI-LEXSUM (SHORT). We give PRIMERA different inputs obtained by using diverse metrics as similarity functions for soft labeling.

Metric	**R-1**$_{f1}$	**R-2**$_{f1}$	**R-L**$_{f1}$
BLEU	43.62	19.18	28.58
R1-F1	44.06	19.33	29.32
R1-P	40.97	16.96	26.51
R2-F1	**45.29**	**20.20**	**30.19**
R2-P	43.32	19.20	29.14
RL-F1	43.18	19.34	28.30
RL-P	43.90	19.14	29.15

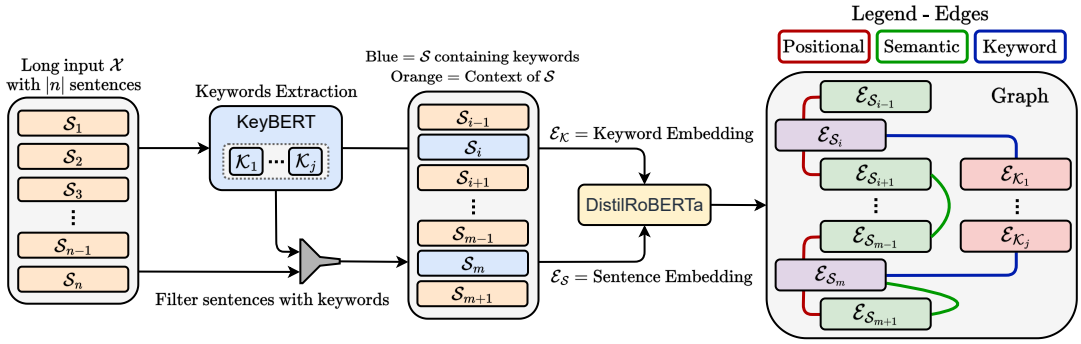


FIGURE 5.2. The pipeline adopted by G-SEEK to produce a semantic heterogeneous graph from a long input. The document is given to KEYBERT, which creates a set of unique keywords. Then (i) the sentences containing at least one keyword, (ii) their context (sentences immediately next or before), (iii) and keywords are turned into embeddings using DISTILROBERTA, becoming the new graph nodes linked with different meaningful edges.

for the target summary to train a model to recognize such salient sentences. Regardless, because of the lack of a ground-truth relevancy label, we need to heuristically mark the salience of the input sentences w.r.t. the gold summary, namely performing a soft labeling strategy. Concretely, let $\mathcal{X} = \{x_1, \ldots, x_x\}$ and $\mathcal{Y} = \{y_1, \ldots, y_y\}$ be the long input and the corresponding summary, respectively, where each $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ is a sentence. For each x_i, we produce a relevancy score $\in [0, 1]$ by computing the similarity between each y_i (and then taking the best value), using a greedy algorithm (Algorithm 6). We test different evaluation metrics (i.e., BLEU [Papineni et al., 2002] and ROUGE Lin [2004]) as similarity functions on MULTI-LEXSUM (SHORT) [Shen et al., 2022], using 100 samples for training and validation sets. We employ PRIMERA [Xiao et al., 2022] as the backbone summarization model, which is a transformer with linear complexity in the input length pre-trained with an MDS-specific objective. Technically, after assigning a summary-relevance score to each $x_i \in \mathcal{X}$, we give PRIMERA only the sentences (in the order of occurrence in the source) with the highest scores until the maximum input size of the model (i.e., 4096 tokens). Table 5.1 indicates that the best metric for soft labeling is ROUGE-2 F1, so we use such a metric to label the relevancy of sentences. We denote that Table 5.1 reports the results by applying the soft labeling also on the validation set to investigate upper-bound performances (i.e., we simulate an oracle by accessing the ground-truth target summaries).

5.1.3.2 Document-to-Graph

The document-to-graph module aims to create a semantic heterogeneous graph with the following steps (Figure 5.2):

- **Keyword Extraction**: we first remove the English stop words and general domain-specific terms that occur in more than 40% of the input (in each MDS document). Then we employ KEYBERT [Grootendorst, 2020, Mathur and Su-

chithra, 2022], a lightweight method compared to more resource-demanding solutions [Nikzad-Khasmakhi et al., 2021], to select up to k keywords. For MDS, we extract k keywords for each document in the cluster and combine these keyword lists into a unique set, dropping duplicates.

- **Sentence Filtering**: we split the input into sentences and select those with at least one keyword. Additionally, we pick n sentences appearing before and after the selected one as context. For MDS, we define the cluster of sources z as $\{[x_1^1, \ldots, x_x^1], \ldots, [x_1^z, \ldots, x_x^z]\}$. Let x_b^1 and x_e^1 be two sentences with keywords. We then select $[x_{b-n}^1, \ldots, x_{b+n}^1, x_{e-n}^1, \ldots, x_{e+n}^1]$, where $< x_{b-n}^1, \ldots, x_{b+n}^1 > -\{x_b\}$ is the context of x_b.

- **Sentence & Keyword Embedding**: we produce the embedding of all keywords and key sentences using DistilRoBERTa [Sanh et al., 2019], a frozen distilled PLM characterized by a few parameters (82M) that let our solution be efficient in terms of GPU memory computation and occupation. Specifically, this model is already pre-trained to create sentence embeddings by using a self-supervised contrastive learning objective.[3] About creating sentence embeddings, the model yields one representation for each token within a sentence. Then, following Reimers and Gurevych [2019], we average the token embeddings using mean pooling, generating the final vector e_i^x.

- **Graph Creation**: all keyword (KE) and sentence embeddings (SE) become the nodes of our graph. Inspired by Wu et al. [2021b], we use KE as supernodes, which means that all sentences that hold a keyword have a bidirectional *Keyword Edge* (blue lines in Figure 5.2) with the keyword node instead of among them. Then, we add bidirectional *Positional Edges* (red lines in Figure 5.2) between two SE if they appear consecutively in the source. Finally, similarly to Li et al. [2020], Chen et al. [2021a], we add *Semantic Edges* (green lines in Figure 5.2) between e_i and e_j if their cosine similarity is greater than a threshold t. The graph has as many nodes as the number of sentences and keywords.

5.1.3.3 Graph Attention Network

We operate a GAT to learn the relationships between the nodes in the graph (*structural information*) along with the information in their nodes (*semantic information*). By considering the edges connecting the nodes, a GAT can learn to propagate information across sentences, better understanding each sentence's context and meaning. In addition, GATs can efficiently handle enormous and complex graphs. In this study, we used the GAT to identify the most critical sentences by assigning an unbounded positive relevancy score to each node.

Technically, this module has the following layers:

[3]Model: https://huggingface.co/sentence-transformers/all-distilroberta-v1.

- **Reprojection Layer** consists of two linear feed-forward layers (FFL) that learn how to reproject (x') the node embeddings x in the vector space. Mechanically, it expands the dimension d_x of the input embeddings n ($\mathbb{R}^{n\times 768}$) of a factor called the *Boom Factor* (BF), inspired by the transformer architecture [Vaswani et al., 2017].

$$x' = \text{FFL}_\sigma(\text{FFL}_\gamma(x, d_x \cdot \text{BF}), d_x) \tag{5.1}$$

  where σ and γ are learnable parameters of different linear layers.

- **GAT Layer** [Velickovic et al., 2018] exploits structural information to enrich node semantics. The output $\widehat{x}$ is generated by interpolating all rows of x' weighed by a score a ($\mathbb{R}^{n\times 768}$).[4]

$$\widehat{x}_{i,j} = a(\mathbf{W}x'_i, \mathbf{W}x'_j) \tag{5.2}$$

- **Scoring Layer** comprises two linear layers to reduce the size of each node ($\widehat{x}$) to a unique real number (s) used as the relevancy score of the sentence associated with the node ($\mathbb{R}^n$).

$$s = \text{FFL}_\beta(\text{FFL}_\theta(\widehat{x}, d_{\widehat{x}} \cdot \text{BF}), 1) \tag{5.3}$$

To supervise the GAT model, we take advantage of the soft labels (l) explained in Section 5.1.3.1. Technically, we train the model to produce scores (s) that minimize the following loss:

$$\mathcal{L}_{\text{mse}} = (s - l)^2 \tag{5.4}$$

5.1.3.4 Summarization Pipeline

Once the input is turned into a graph and the GAT module assigns a score to each sentence, we extract the most relevant and use a generative PLM to generate the output summary. Technically, according to their relevancy scores, we select the most salient sentences and create an input text for the model containing fewer tokens than its maximum input size. As a result, the new input comprises sentences in the order of occurrence in the raw textual source.

We train the summarizer using the standard cross-entropy loss, which requires the model to predict the next token w_i of the target $\mathcal{Y}$ given $\mathcal{X}$ and the previous target tokens $w_{1:i-1}$, as follows:

$$\mathcal{L}_{\text{ce}} = -\sum_{i=1}^{|\mathcal{Y}|} \log p_\tau(w_i | w_{1:i-1}, \mathcal{X}) \tag{5.5}$$

where τ indicates the model parameters and p is the predicted probability over the vocabulary.

Note: G-Seek *is not jointly trained with the summarization model, resulting in a non-overwhelming training process. Thus, the pipeline of our overall solution is designed to work with a few labeled samples.*

[4] Please refer to the original paper [Velickovic et al., 2018] for further information.

TABLE 5.2. Statistics of the evaluation datasets including size, number of source documents per instance, number of total words in source and target texts, and source-target coverage, density, and compression ratio of words [Grusky et al., 2018]. Except for the number of samples, all reported values are averaged across all instances.

		Source			Target		Source → Target		
Dataset	**Samples**	Docs	Words	Sents	Words	Sents	Coverage	Density	Compress
MDS (Multi-LexSum)									
TINY	1603	10.7	119072.6	5962.5	24.7	1.4	0.92	2.27	5449.6
SHORT	3138	10.3	99378.2	5017.0	130.2	5.1	0.96	3.33	840.7
LONG	4534	8.8	75543.2	3814.2	646.5	28.8	0.94	4.07	97.4
LDS									
GOVREPORT	19,463	1	8765.0	298.7	556.3	18.1	0.94	9.08	17.9

5.1.4 Experiments

We focus our investigation on low-resource summarization, a real-world scenario distinguished by the scarcity of data due to the high cost of labeling. Following previous work [Chen and Shuai, 2021, Moro and Ragazzi, 2022], we take the first 100/10/100 samples from all datasets' training, validation, and test sets without performing additional data pre-processing.

5.1.4.1 Experimental Setup

Datasets We contemplate the **Multi-LexSum** dataset [Shen et al., 2022] as the evaluation benchmark for MDS, which gathers real-world federal civil rights lawsuits with expert-authored summaries. The main challenge of MULTI-LEXSUM is the long size of the source documents and the different granularity of the summaries (i.e., tiny, short, and long). Due to this multi-target nature, we overall experiment with three distinct dataset renditions as testbeds. Regarding LDS, we consider **GovReport** [Huang et al., 2021], comprising 19K U.S. government reports. Table 5.2 provides key measurements of the datasets.[5]

Baselines We compare with cutting-edge abstractive summarization models. **Bart** [Lewis et al., 2020a] is a transformer with a quadratic memory and time complexity in the input length; we use the base and large checkpoints. **Pegasus** [Zhang et al., 2020a] is a quadratic transformer pre-trained with a summarization-specific objective to predict gap sentences as a pseudo summary; we utilize the large checkpoint. **Led** [Beltagy et al., 2020] is a transformer with a linear memory complexity thanks to a sparse attention mechanism; we employ the large checkpoint. **Primera** [Xiao et al., 2022] is a linear transformer with the same architecture as LED but with an MDS-specific pretraining objective to generate pseudo summaries, which are text spans automatically extracted

[5]All datasets are publicly available in Hugging Face: https://huggingface.co/datasets/allenai/multi_lexsum (MULTI-LEXSUM) and https://huggingface.co/datasets/ccdv/govreport-summarization (GOVREPORT).

TABLE 5.3. The number of trainable parameters of generative PLMs and their maximum input size. Each URL starts with "https://huggingface.co/." G-SEEK uses the max input length of the downstream model but provides salient sentences instead of truncating the exceeding ones.

	URL	#Params	Input
Models			
BART-B	facebook/bart-base	140M	1024
BART-L	facebook/bart-large	400M	1024
PEGASUS-L	google/pegasus-large	568M	1024
LED-L	allenai/led-large-16384	459M	4096
PRIMERA-L	allenai/PRIMERA	447M	4096
G-SEEK	-	4M	-

TABLE 5.4. Evaluation F1 scores on the benchmarked MULTI-LEXSUM (TINY, SHORT, LONG) and GOVREPORT datasets. B and L denote base and large, respectively. The best score for each model is in bold. † means statistically significant results of G-SEEK (p-value < 0.05 with student t-test).

	Multi-LexSum (Tiny)					Multi-LexSum (Short)					Multi-LexSum (Long)					GovReport				
Model	**R-1**	**R-2**	**R-L**	$\mathcal{R}$	**BS**	**R-1**	**R-2**	**R-L**	$\mathcal{R}$	**BS**	**R-1**	**R-2**	**R-L**	$\mathcal{R}$	**BS**	**R-1**	**R-2**	**R-L**	$\mathcal{R}$	**BS**
Quadratic																				
BART-B	18.49	**6.39**	16.10	13.62	72.70	39.20	**17.52**	35.20	**30.37**	79.41	41.22	15.53	39.12	31.53	78.72	47.26	13.58	44.27	34.24	80.61
w/ G-SEEK	**20.85**†	5.82	**17.14**†	**14.54**†	**74.20**†	**40.45**†	15.34	**35.31**	30.01	**79.84**†	**42.28**†	**16.36**†	**40.08**†	**32.46**†	**79.62**†	**47.65**†	**14.96**†	**44.38**	**34.91**†	**81.18**†
BART-L	22.37	**7.91**	19.74	16.61	76.17	41.45	**18.74**	**35.81**	**31.70**	79.89	41.41	16.47	38.98	31.88	79.13	48.46	14.03	44.83	34.94	80.82
w/ G-SEEK	**24.46**†	7.70	**20.34**†	**17.41**†	**77.07**†	**41.57**	16.72	35.78	31.01	**79.97**	**43.92**†	**17.43**†	**41.08**†	**33.67**†	**80.19**†	**51.46**†	**17.12**†	**48.05**†	**37.97**†	**81.78**†
PEGASUS-L	15.09	3.20	12.07	10.09	70.27	38.29	16.31	32.63	28.83	78.58	40.19	16.13	37.82	31.02	78.39	47.12	14.07	44.82	34.55	80.51
w/ G-SEEK	**19.84**†	**5.13**†	**16.28**†	**13.70**†	**73.12**†	**38.78**†	**16.32**	**33.26**†	**29.18**†	**79.11**†	**42.38**†	**17.04**†	**39.88**†	**32.68**†	**79.53**†	**50.55**†	**17.06**†	**48.01**†	**37.67**†	**81.30**†
Linear																				
LED-L	22.86	**7.98**	18.86	16.50	76.20	40.09	**17.50**	35.15	**30.63**	79.51	45.26	**20.01**	42.66	**35.52**	**81.31**	53.86	19.53	49.28	39.96	82.65
w/ G-SEEK	**24.39**†	7.96	**20.55**†	**17.55**†	**77.19**†	**40.95**†	16.28	**35.31**	30.51	**80.56**†	**45.42**	18.87	**42.93**†	35.23	81.03	**55.63**†	**21.08**†	**50.67**†	**41.49**†	**82.77**†
PRIMERA-L	25.37	**8.13**	20.84	18.02	76.45	40.20	14.88	34.88	29.63	80.31	45.31	**21.06**	42.44	**35.85**	81.34	54.20	19.37	50.20	40.28	79.75
w/ G-SEEK	**25.76**†	7.59	**21.36**†	**18.13**	**77.26**†	**43.99**†	**18.67**†	**37.55**†	**33.02**†	**81.32**†	**45.92**†	19.61	**42.59**	35.55	**81.36**	**57.13**†	**21.20**†	**53.64**†	**42.87**†	**80.37**†

based on the entity salience; we use the large checkpoint. Technically, in MDS, we use the standard approach of concatenating documents from the same cluster to form a single long textual input (we add the special token `<doc-sep>` to separate documents [Xiao et al., 2022]). Note: we denote the base and large versions by B and L, respectively. Table 5.3 reports the number of parameters and maximum input size of the models.[6]

Metrics We embrace the conventional ROUGE-{1,2,L}[7] F1 [Lin, 2004] and BERTScore F1 (BS) [Zhang et al., 2020b] to calculate the syntactic and semantic overlap, respectively, between the inferred and ground truth summaries. Moreover, we tally $\mathcal{R} = {}^{\text{avg}(r_1,r_2,r_L)}/_{1+\sigma_r^2}$ [Moro et al., 2023b] to also consider an aggregated judgment, where σ_r^2 is the variance of the average ROUGE scores that penalizes generations with heterogeneous results across dimensions. All metrics $\in [0, 1]$ (%); the higher, the better.

Implementation Details We fine-tune the models based on the PyTorch [Paszke et al., 2019] implementations of the HuggingFace library [Wolf et al., 2019], setting

[6]The maximum input length depends on the architecture of the models' encoder, whereas the output size hinges on the dataset.

[7]We use the summary-level R-L, where each summary is split into sentences.

the seed to 42 for reproducibility. All experiments are run on an internal workstation with a Nvidia RTX 3090 GPU of 24 GB memory, 64 GB of RAM, and an Intel(R) Core(TM) i9-10900X CPU @3.70GHz processor. Regarding the GAT module of G-Seek, the training lasted 75 epochs with a learning rate of 5e-5, using AdamW as the optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$.[8] About the summarization task, all models are trained for 5 epochs with a learning rate of 3e-5, using mixed precision and gradient checkpointing to preserve memory. For decoding, we operate beam search with 5 beams and n-gram repetition blocks for n>5, utilizing the following (min-max) summary length: Multi-LexSum's Tiny (10-50), Short (50-150), Long (350-750), and GovReport (500-1000).

5.1.4.2 Overall Results

We train and evaluate all models on the benchmark datasets with and without G-Seek to highlight our contribution. Table 5.4 reports the performance of the systems in MDS and LDS. In particular, G-Seek consistently improves model performance across datasets and metrics, showing an ameliorative contribution of our method that gives only salient information to generative PLMs.

5.1.4.3 Graph — Experiments

We investigate different graph settings. In particular, we focus on the *Sentence Filtering* module, examining the precision and recall of all labeled sentences among the selected salient ones. Table 5.5 reports the results, experimenting over the validation set of Multi-LexSum (Short) with the following facets:

- **Keywords**. We study the maximum number of keywords extracted by KeyBERT for each document in the cluster, establishing that 5 achieves the best results.
- **Context**. We examine a different number of consecutive sequences selected as the context of the salient sentence, discovering that 4 leads to better results.

With our hardware, the average time to create the graph for a single long input of ≈ 100K words is about 34 seconds. We denote that the current implementation does not adopt a specific optimization.

5.1.4.4 GAT — Experiments

We experiment with our GAT module under various settings in the validation set of Multi-LexSum (Short). Specifically, we train the GAT for 30 epochs and evaluate by picking the top 100 sentences according to the assigned score. We use these 100 items to compute precision, recall, and f-measure. Table 5.6 reports the results, where we test the following:

[8] Since all evaluation corpora are legal datasets, we trained the GAT module once on Multi-LexSum (Short) using soft labels (Section 5.1.3.1).

TABLE 5.5. Results of different graph settings on the validation set of MULTI-LEXSUM (SHORT) evaluated with precision, recall, and f-measure.

	P	R	F1
Keywords			
4	22.12	76.43	34.31
5	**23.29**	**78.81**	**35.95**
6	22.98	75.42	35.22
7	22.65	77.26	35.03
8	22.57	77.12	34.92

	P	R	F1
Consecutive Sentences			
1	**23.26**	62.64	33.92
2	22.12	76.43	34.31
3	23.24	76.00	35.60
4	23.15	**78.85**	**35.79**
5	23.06	77.42	35.54

TABLE 5.6. The results of the GAT module on the validation set of MULTI-LEXSUM (SHORT) under different settings with 30 training epochs. The final GAT is the best setting and checkpoint after 75 epochs.

	P	R	F1
Boom Factor			
1	31.34	38.43	34.52
2	**32.12**	**39.17**	**35.29**
3	31.69	38.80	34.88
4	31.52	38.54	34.68
GAT Layers			
1	**35.38**	**42.31**	**38.54**
2	32.13	39.18	35.31
3	28.34	35.13	31.37
4	29.45	36.37	32.55

	P	R	F1
Cosine Similarity			
0.80	32.02	39.01	35.17
0.82	32.10	39.18	35.28
0.84	32.10	39.17	35.28
0.86	**32.49**	**39.55**	**35.67**
0.88	31.96	39.01	35.13
Final GAT			
	38.37	45.49	41.63

- **Boom Factor**. We explore the contribution of the Boom Factor in the *Reprojection Layer*, finding that 2 is the best value.
- **Layers**. We employ various layers, surprisingly uncovering that 1 is the best option. Therefore, the small pool of training examples favors a lightweight solution with few trainable parameters.
- **Cosine Similarity**. We investigate the threshold to create *Semantic Edges* between nodes, deciding that 0.86 is the best choice.

Table 5.6 further shows the results of an optimal GAT trained for 100 epochs. We then select the best model checkpoint, which is after 75 epochs. The average time for each epoch is about 60 s.

Note: after numerous experiments, we tested the module by altering each hyperparameter while keeping the other two constants, using their best values (highlighted in bold in Table 5.6).

TABLE 5.7. A random qualitative input example of G-SEEK against a truncation-based approach on MULTI-LEXSUM (SHORT).

Golden Summary
Pursuant to the Civil Rights of Institutionalized Persons Act ("CRIPA"), 42 U.S.C. § 1997, the Civil Rights Division of the U.S. Department of Justice ("DOJ") conducted an investigation of conditions at the Mercer County Geriatric Center ("MCGC"), a public nursing home facility in New Jersey, evidently operated by Mercer County. The investigation led the DOJ to find that certain conditions at MCGC violated residents federal rights. The parties settled and the case is now closed.
Truncation-based Input
[... Text ...] The Attorney General files this complaint on behalf of the United States of America pursuant to the Civil Rights of Institutionalized Persons Act, 42 U.S.C. § 1997, to enjoin the named Defendants from depriving residents housed in the Mercer County Geriatric Center (MCGC) of rights, privileges, or immunities secured and protected by the Constitution and laws of the United States. *[... Text ...]*
G-Seek Input
Administrator ECROYD is sued in his official capacity. 11. Mercer County receives federal Medicare and Medicaid funds for care provided at MCGC. 15.Defendants and MCGC are public entit(ies)" under the ADA and implementing regulations. 18.Defendant MERCER COUNTY is the entity charged by the laws of the State of New Jersey with authority to operate the MCGC and is responsible for the living conditions and health and safety of persons living in MCGC. 8. Through their acts and omissions, Defendants have failed to provide "care for its residents in such matter and in such an environment as will promote maintenance or enhancement of the quality of life of each resident, and have further failed to provide "the necessary care and services to attain or maintain the highest practicable physical, mental, and psychosocial well. *[... Text ...]* If that resident was a candidate for services at home or in another community setting, this failure to accommodate a disability effectively resulted in the lengthy and improper segregation of the resident from society.

5.1.4.5 Analysis on Input Quality

Table 5.7 shows a qualitative example in MULTI-LEXSUM (SHORT) of our G-SEEK approach with respect to the input to give to summarization models. Table 5.8 reveals the higher source-target correlation using G-SEEK that creates an input composed of more key sentences. Specifically, we compare our method to the standard input truncation approach on the GOVREPORT's test set by computing ROUGE, BERTScore (also providing precision and recall scores), and unique keyword occurrence (dubbed UKO) between the inputs to give to models and their corresponding gold summaries. About UKO, we count the % of unique keywords (i.e., all words that are not stopwords) in the target that also appear in the source.

TABLE 5.8. The correlation between source-target pairs on the test set of the GOVREPORT dataset.

Approach	R-1$_{f1}$	R-2$_{f1}$	R-L$_{f1}$	BS$_p$	BS$_r$	BS$_{f1}$	UKO
TRUNCATION	31.47	11.73	30.28	69.47	72.24	70.81	67
G-SEEK	**33.42**	**12.88**	**33.19**	**73.93**	**75.86**	**74.87**	**71**

5.1.5 Conclusion

We propose G-SEEK, a novel graph-based method to extract and provide essential knowledge from massive textual information to abstractive summarization models for synthesis generation. G-SEEK represents a long input with a heterogeneous graph and models its semantics through different units (i.e., sentences and keywords) to pinpoint only the salient sentences. Experimental results in a data scarcity scenario over multiple public LDS and MDS datasets show that G-SEEK significantly improves the performance of syntactic and semantic metrics of state-of-the-art summarization systems. Furthermore, we show the contribution of G-SEEK to yield more correlated source-target pairs that enable generative PLMs to learn faster with few labeled training instances.

Future works should explore lightweight end-to-end pipelines (to train the GAT with the generative PLM jointly) and the contribution of the number of salient sentences to the final summary generation. Furthermore, as presented for communication networks [Lodi et al., 2010, Cerroni et al., 2013, 2015], tracking and propagating knowledge refinements in sentences could be crucial when modeling long texts with graphs.

Ethics Statement

Because of the high societal impact of the legislation and biases in current generative PLMs [Liang et al., 2021], experts should determine the quality of the predicted summaries to make the presented solution work in real-world applications. All datasets and models used in this study were already publicly available.

5.2
REVELIO: Interpretable Long-Form Question Answering

Originally published as: *Gianluca Moro, Luca Ragazzi, Lorenzo Valgimigli, Fabian Vincenzi, and Davide Freddi.* REVELIO*: Interpretable long-form question answering. In The Second Tiny Papers Track at ICLR 2024, Tiny Papers @ ICLR 2024, Vienna, Austria, May 7-11, 2024. OpenReview.net, 2024*

Abstract

The black-box architecture of pretrained language models (PLMs) hinders the interpretability of lengthy responses in long-form question answering (LFQA). Prior studies use knowledge graphs (KGs) to enhance output transparency, but mostly focus on non-generative or short-form QA. We present REVELIO, a new layer that maps PLM's inner working onto a KG walk. Tests on two LFQA datasets show that REVELIO supports PLM-generated answers with reasoning paths presented as rationales while retaining performance and time akin to their vanilla counterparts.

5.2.1 Introduction

Closed-book long-form question answering (LFQA) asks pretrained language models (PLMs) to generate long responses using only the knowledge stored in their parameters. While avoiding open-book passage retrieval accelerates training and inference, the black-box nature of PLMs hampers answer interpretation, limiting real-world applicability. To improve human understanding [Commission, 2020], prior studies use knowledge graphs (KGs) to provide logical reasoning [Yasunaga et al., 2021, Zhang et al., 2022c]. However, they focus on non-generative multiple-choice QA (MCQA), neglecting the practicality of LFQA (see Appendix 5.2.A). We introduce REVELIO, a plugin layer that enables PLMs to craft answers by aligning parametric knowledge with an external KG, offering reasoning paths as support for decision interpretation. Our study delves into two closed-book LFQA datasets by testing two million-scale PLMs. Quantitative and qualitative results show that REVELIO provides reasoning pathways as the rationale behind a given answer while marginally improving performance. Our code is open at `https://disi-unibo-nlp.github.io/projects/revelio/`.

5.2.2 Method

We introduce REVELIO, our proposed plugin layer to allow PLMs to communicate with an external KG to provide answer-related reasoning paths to enhance interpretability (Figure 5.3).

Preprocessing. Given a question x, we use RAKE [Rose et al., 2010] to extract a set of keywords $\mathcal{K} = \{k_1, \ldots, k_{|\mathcal{K}|}\}$. We create $|\mathcal{K}|$ depth-d subgraphs $\mathcal{G} = \{g_1, \ldots, g_{|\mathcal{K}|} \mid$

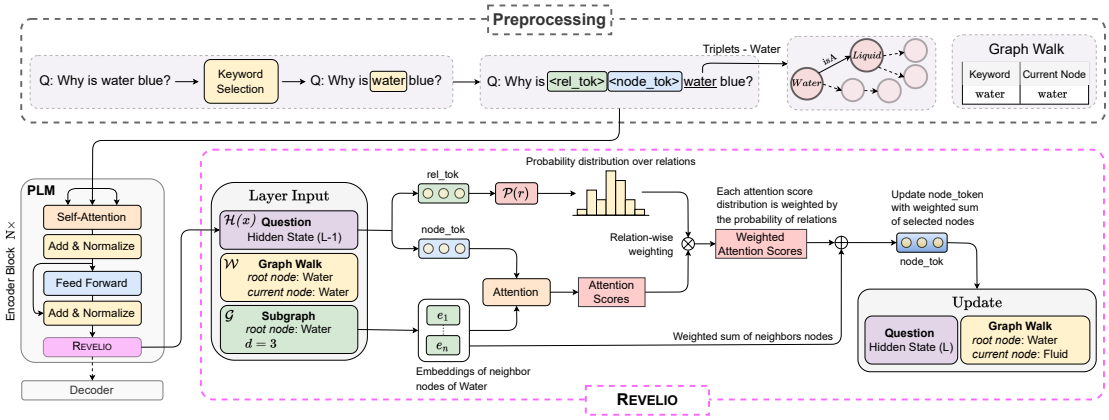


FIGURE 5.3. The overview of our proposed approach.

$g_i = \langle \mathcal{N}_r, \mathcal{E}, \mathcal{N}_e \rangle\}$, setting k_i as the root node of g_i, where d is a hyperparameter and $\langle \mathcal{N}_r$, $\mathcal{E}$, $\mathcal{N}_e \rangle$ are relational triplets with $\mathcal{N}_r$, $\mathcal{N}_e$, and $\mathcal{E}$ representing the root node, end node, and edges, respectively. For each k, we perform an exact match on CONCEPTNET [Speer et al., 2017] to retrieve the corresponding depth-d subgraph starting from k. Following Hu et al. [2022b], we augment x by preceding each k with new tokens `<rel_tok>` and `<node_tok>` to allow PLMs to interact with $\mathcal{E}$ and $\mathcal{N}$, respectively. We then define the REVELIO graph walk $\mathcal{W} = \{w_1, \ldots, w_{|\mathcal{K}|}\}$, where w_i is the path on g_i starting from k_i. Iteratively, REVELIO adds a triplet (e.g., $\langle$*water*, *is-a*, *liquid*$\rangle$) to each w_i, symbolizing a step in the walk. For simplicity, we define the end node of the last triplet as *Current Node*, representing the position of the model in the graph. Appendix 5.2.C reports in-depth elucidation and ablation studies.

Execution. For each w_i, REVELIO aims to select the next most salient node and inject the new information from $\mathcal{G}$ into $\mathcal{H}_x$—the last hidden state of x. To streamline, we explain the process for a single k, but the same mechanism is run in parallel for each $k \in \mathcal{K}$. The input of REVELIO is $(\mathcal{H}_x, \mathcal{G}, \mathcal{W})$. First, we extract e_r, e_n from $\mathcal{H}_x$, i.e., the `<rel_tok>` and `<node_tok>` embeddings, respectively. We then linearly project e_r and apply a softmax operation to yield a probability distribution $\mathcal{P}(r)$ over all possible edges (24 types in CONCEPTNET, e.g., *is-a*, *is-composed-of*). Meanwhile, we perform an attention operation to compare e_n with all neighbor nodes n of the *Current Node* of w, producing a score s_e for each pair. The scores are then weighted by $\mathcal{P}(r)$, considering the relation that links the *Current Node* and n. The node with the highest score is added to $\mathcal{W}$. Then, all node embeddings, weighted by their scores, are summed to e_n to inject KG information into the PLM.

5.2.3 Experiments

Setup. We train and evaluate the two most popular million-scale PLMs, such as BART [Lewis et al., 2020a] and T5 [Raffel et al., 2020], with and without REVELIO on two closed-book public LFQA datasets: ELI5 [Fan et al., 2019] and AQUAMUSE [Kulka-

Model (size)	Eli5						AQuaMuSe					
	R1	R2	RL	$\mathcal{R}$	BeS	BaS	R1	R2	RL	$\mathcal{R}$	BeS	BaS
T5 (base)	22.93	4.11	14.02	13.61	2.91	-2.48	30.09	7.98	19.67	19.09	**11.61**	**-2.29**
w/ REVELIO	**23.01**	**4.26**	**14.17**	**13.73**	**3.10**	**-2.47**	**30.14**	**8.18**	**20.42**	**19.42**	11.05	-2.43
T5 (large)	23.35	4.21	**14.25**	13.85	**5.35**	-2.31	**30.87**	8.95	**20.74**	**20.03**	**17.52**	-2.17
w/ REVELIO	**23.57**	**4.26**	14.16	**13.91**	5.06	**-2.30**	30.62	**9.17**	20.45	19.93	15.58	**-2.11**
BART (base)	23.21	3.33	13.16	13.15	-15.32	-5.40	24.11	4.72	17.14	15.23	-11.72	-5.41
w/ REVELIO	**23.41**	**3.74**	**13.27**	**13.39**	**-14.22**	**-5.27**	**24.67**	**4.78**	**17.32**	**15.49**	**-10.98**	**-5.21**
BART (large)	24.23	4.17	14.71	14.27	-7.54	-3.06	**26.67**	5.98	**19.52**	**17.26**	-5.28	**-3.41**
w/ REVELIO	**24.62**	**4.23**	**15.12**	**14.56**	**-7.22**	**-2.58**	26.53	**6.12**	19.41	17.23	**-5.11**	-3.59

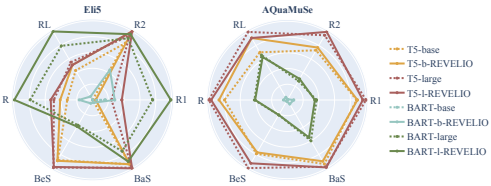


FIGURE 5.4. Results on the benchmarked datasets. The best intra-model score in the table is in bold.

rni et al., 2020]. For all benchmarks, we automatically evaluate models by reporting % of the F1 scores of ROUGE-{1,2,L} [Lin, 2004] and $\mathcal{R}$ [Moro et al., 2023b] for syntactic matching, BERTScore (BeS) [Zhang et al., 2020b] for semantic assessment, and BARTScore-$\mathcal{F}$ (BaS) [Yuan et al., 2021] to judge factuality. We perform human analysis using a direct comparison strategy that has been proven to be more reliable and less labor-intensive than rating scales [Huang et al., 2023, Moro et al., 2023d]. More details are in Appendix 5.2.B.

Results. Figure 5.4 shows the overall results of REVELIO. In detail, the adoption of REVELIO overall slightly improves model performance across datasets and metrics in two different LFQA tasks. This finding indicates the benefit of interacting with KGs to extract contextualized information. Human annotation (with an agreement of 78%) shows that 85% of the time REVELIO's answers are comparable or better than those of T5 (see Appendix 5.2.B for details). Ablation studies are given in Appendix 5.2.C. Finally, graphical examples of reasoning paths are provided in Appendix 5.2.D.

5.2.4 Conclusion

We present REVELIO, a flexible layer to enhance the output of current PLMs with interpretable reasoning paths. Experiments on two closed-book LFQA datasets show that models equipped with REVELIO generate better answers than their vanilla counterparts, paving the way for novel promising KG interaction methods for LFQA. Limitations and future directions are discussed in Appendix 5.2.E.

URM Statement

The authors acknowledge that at least one key author of this work meets the URM criteria of ICLR 2024 Tiny Papers Track.

5.2.A Related Work

Currently, there is strong interest in explaining the reasoning behind the output of PLMs to enhance interpretability [Frisoni and Moro, 2020]. To this end, the synergy between KGs and PLMs has been explored in the field of QA. QA-GNN [Yasunaga et al., 2021] extracts subgraphs for each possible answer, and a multi-layer perceptron assigns the probability of being the correct answer. GREASE-LM [Zhang et al.,

TABLE 5.9. Related works about the integration of QA and KG.

Model	Task	KG*	Plug†	Gen‡	PLM (Size)
QA-GNN [Yasunaga et al., 2021]	MCQA	ConceptNet, UMLS, DrugBan	✗	✗	RoBERTa (large), AristoRoBERTa
GreaseLM [Zhang et al., 2022c]	MCQA	ConceptNet, Desease DB	✗	✗	RoBERTa (large), SapBERT
CORN [Guan et al., 2022]	MCQA	ConceptNet	✗	✗	RoBERTa (large)
ACENet [Hao et al., 2022]	MCQA	ConceptNet	✗	✗	RoBERTa (large), AristoRoBERTa, SapBERT
DRKL [Zhang et al., 2022b]	MCQA	ConceptNet, Desease DB	✗	✗	RoBERTa (large), SapBERT
Dragon [Yasunaga et al., 2022]	MCQA	Freebase, Wikidata, ConceptNet	✗	✓	RoBERTa (large)
GrapeQA [Taunk et al., 2023]	MCQA	ConceptNet	✗	✗	RoBERTa (large), SapBERT
SAFE [Jiang et al., 2022]	MCQA	ConceptNet	✓	✗	RoBERTa (large), AristoRoBERTa, BERT(large)
OreoLM [Hu et al., 2022b]	MCQA	ConceptNet	✓	✓	RoBERTa, T5(base/large)
Revelio *(Ours)*	LFQA	ConceptNet	✓	✓	T5 (base/large), Bart (base/large),

* The external knowledge graphs used.
† ✓ = the solution can be plugged into different PLMs with minimal effort and the paper contains experiments on that; ✗ = otherwise.
‡ ✓ = the output is generated (i.e., abstractive); ✗ = otherwise.

2022c] extends the work by creating connections between KG and PLM. Other solutions, such as Corn [Guan et al., 2022] and AceNet [Hao et al., 2022], improve graph processing with PLM information and vice versa using different attention mechanisms. SAFE [Jiang et al., 2022] simplifies the process by considering only relations, not nodes, proving that the relationships are sufficient for this task. The paradigm change is given by Dragon [Yasunaga et al., 2022], a generative model pretrained by coupling masked language modeling and link prediction, outperforming all previous models in the QA task. So far, all models work by generating a probability over the possible answers. Therefore, the applicability is narrowed to answer selection, i.e., MCQA, leaving generative QA unexplored. OreoLM [Hu et al., 2022b] tries to mitigate this problem using a generative model to produce a short answer. However, all these existing solutions focus on MCQA, with the aim of guessing the most probable answer given a set of alternatives or generating an answer composed of a few words. In this work, we address the generation of long and complex answers, leveraging a KG to improve model performance and interpret the reasoning. Table 5.9 compares our contribution with previous works.

5.2.B Experimental Details

Models. Bart is a transformer with quadratic memory and time complexity in input size characterized by a denoising pretraining objective. T5 is a quadratic transformer with a text-to-text objective. Table 5.10 lists the number of parameters and the URL of the model checkpoints.

Datasets. Eli5 comprises question–answer pairs extracted from the Reddit forum

TABLE 5.10. The number of trainable parameters of PLMs.

Model	Parameters	URL
BART-base	140M	https://huggingface.co/facebook/bart-base
BART-large	400M	https://huggingface.co/facebook/bart-large
T5-base	220M	https://huggingface.co/t5-base
T5-large	770M	https://huggingface.co/t5-large
REVELIO	2M	-

TABLE 5.11. Statistics of the datasets used as testbeds. All values are averaged except "# Instances."

					Source	Target
Dataset	**Domain**	**# Train**	**# Dev**	**# Test**	# Words	# Words
ELI5 [2019]	Commonsense	5000	100	1000	42.2	130.6
AQUAMUSE [2020]	Commonsense	4555	440	524	15.5	105.9

"Explain Like I'm Five." AQUAMUSE collects query-based summaries of topic-related documents. In our experiments, given the input question, we directly use the summary as the answer without providing the source documents to the models. Table 5.11 provides the dataset statistics.[9]

Metrics. Table 6.55 shows technical details on the automatic evaluation metrics used. Regarding human annotation, we provide three English-proficient QA researchers with 40 random test set questions from ELI5 and their corresponding gold and machine-generated answers from T5-base. We ask the annotators to state which answer is the best in terms of correctness and informativeness w.r.t. the given question. A tie is declared if the judge perceives that the two answers are of comparable quality. We randomize the order of instances to guard the rating against being gamed. We collect the responses and aggregate them using majority voting. The results are depicted in Figure 6.29.

Implementation Details. We fine-tuned the models using PyTorch and the HuggingFace library, setting the seed to 42 to guarantee reproducibility. All PLMs are trained for 5 epochs with a learning rate of $3e^{-5}$, using mixed precision and gradient checkpointing to preserve memory. We selected the checkpoint that performed the best on the validation set at the end of each training epoch. Our REVELIO layer uses a different optimizer whose learning rate is $1e^{-6}$ (see Appendix 5.2.C for particulars). At inference time, we use the beam search decoding with 3 beams, n-gram repetition blocks for n>3, and an output length in $[50, 140]$ tokens.

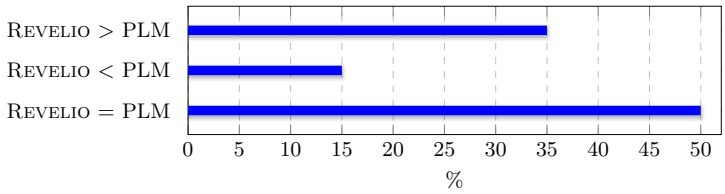
Hardware Configuration. We used a workstation with 4 Nvidia GeForce RTX3090 GPU of 24 GB memory, 64 GB of VRAM, and an Intel® Core™ i9-10900X1080 CPU @ 3.70GHz processor.

[9]All datasets are publicly available in Hugging Face: https://huggingface.co/datasets/eli5 and https://huggingface.co/datasets/aquamuse.

TABLE 5.12. The settings of the used metrics.

Metric	Description	Bound*	Hyperparameters
ROUGE-1/2/L [Lin, 2004]	Lexical overlaps of unigrams (R-1), bigrams (R-2), and longest common subsequence (R-L).	$[0, 1], \uparrow$	rouge_types=["r1","r2","rL"], use_stemmer=True
$\mathcal{R}$ [Moro et al., 2023b]	Aggregated ROUGE score penalizing results with discrepant R-1, R-2, R-L.	$[0, 1], \uparrow$	/
BERTScore [Zhang et al., 2020b]	IDF-weighted n-gram alignment through contextualized embeddings from BERT [Devlin et al., 2019].	$[-1, 1], \uparrow$	model_type="microsoft/deberta-large-mnli", rescale_with_baseline=True, batch_size=32
BARTScore-$\mathcal{F}$ [Yuan et al., 2021]	Estimation of BART of how predictions and references are mutual paraphrases.	$[-\infty, 0], \uparrow$	checkpoint="facebook/bart-large-cnn", batch_size=4

* $\uparrow$ = the higher, the better.



Comparison	Score
REVELIO > PLM	**35%**
REVELIO < PLM	15%
REVELIO = PLM	50%

FIGURE 5.5. Human evaluation results on 40 random samples of ELI5.

TABLE 5.13. ROUGE scores with different hyperparameters. *Left*: a comparison of the number of REVELIO layers (mean of up to 10 different settings). *Right*: different keyword extraction methods.

# Layers	R1	R2	RL
1	20.65	2.81	10.32
2	**21.83**	3.45	12.71
3	21.81	**3.59**	**12.78**
4	20.93	3.06	12.21

Method	R1	R2	RL
RAKE	21.64	**3.71**	**13.09**
JAKE	21.23	3.32	12.89
KEYBERT	20.79	3.02	11.97
YAKE	**21.67**	3.44	12.76

5.2.C Ablation Studies

To optimize hyperparameter selection, we conducted an extensive series of experiments using OPTUNA,[10] a sophisticated open-source tool designed for hyperparameter optimization that inherently supports parallelism. We configure OPTUNA to seamlessly integrate with SLURM, our resource management system. This configuration allowed us to run parallel experiments efficiently on four GPUs, while collecting all the data on a MySQL server located on a separate node for optimized data handling. Table 5.13 reports the results of the ablation studies, performed with T5-base on ELI5.

Learning Rate Optimization. We started by identifying the optimal learning rate for training the PLM with REVELIO. We differentiated between two distinct learning rates: $lr_{\mathcal{R}}$ for custom layers and $lr_{\mathcal{P}}$ for the standard parameters of the model. Initially,

[10] https://optuna.org/

TABLE 5.14. Erroneous output examples from ELI5.

#1 Question
How come we can not feel the heat of the inner earth if we are only 18 miles above it?
#1 Gold Answer
For the same reason we don't burn to death when a steel forge is fired up from a mile away. The asthenosphere's temperature is between 300 and 500 degrees Celsius. The earth's crust, from the surface to the asthenosphere, is 30 to 50 miles thick. What's more, the earth's crust has an extremely low thermal conductivity. No material exists that has a remotely sufficient specific heat capacity to transfer 500C of heat through 18 miles of (highly insulating) earth.
#1 Generated Answer
The heat of the inner earth is absorbed by the air. It's not like we can't feel it, but it's just that we're only 18 miles above it. If you're in a high altitude, you can feel it.

#2 Question
God is a woman ariana grande writing credits?
#2 Gold Answer
"God Is a Woman" (stylized "God is a woman") is a song by American singer Ariana Grande. It was released on July 13, 2018, as the second single from Grande's fourth studio album Sweetener (2018). The song was written by Grande, Max Martin, Savan Kotecha, Rickard Göransson and its producer Ilya.
#2 Generated Answer
God is a woman, a writer, and an actress. She is best known for her role in the television series The Greatest Showman, which premiered on November 5, 2018. She is also known for playing the role of a female character in the TV series The Big Bang Theory.

we considered employing separate optimizers for each set of parameters; however, empirical evidence suggested that it did not provide a significant advantage. Consequently, we adopt a unified optimization approach. Through rigorous testing, we determined that

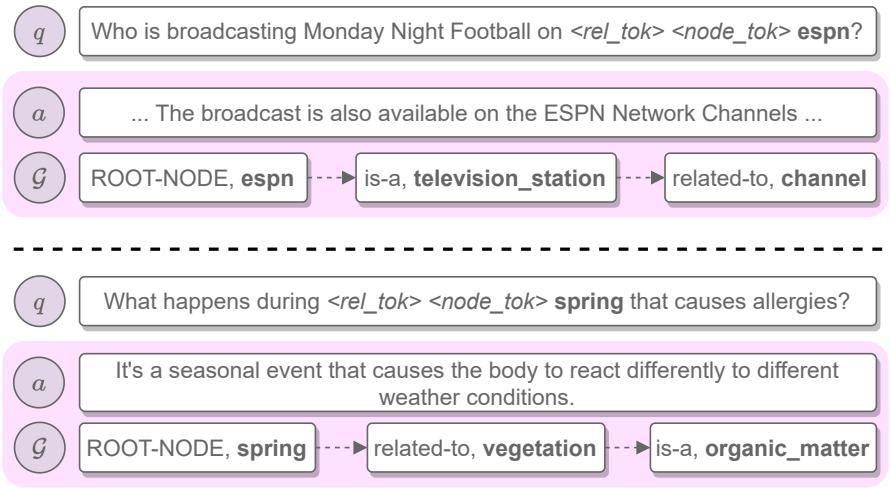


FIGURE 5.6. Graphical examples of reasoning paths from ELI5.

the most effective training occurred with $lr_{\mathcal{R}} = 1e^{-6}$ and $lr_{\mathcal{P}} = 3e^{-5}$.

Revelio Layers. We experimented using REVELIO in different layers. We let OPTUNA decide how many layers to use from 1 to 4, and which layers to alter. Our findings revealed that the incorporation of 2 or 3 REVELIO layers yielded comparable effective results. Given this equivalence in performance, we opt for 2 layers to minimize computational complexity and reduce the depth of the graph $\mathcal{G}$ by 1 layer, thus streamlining the process. Subsequently, we conducted a more focused experiment, fixing the layer count at 2 while using OPTUNA to identify the most impactful layer positions. The most effective configuration emerged as a combination of the 3$^{\text{rd}}$ and 7$^{\text{th}}$ layers, striking a balance between the early and later stages of information processing within the PLM.

Keyword Extraction Algorithm. Another key component investigated in the early phase was the keyword extractor. We tested RAKE, JAKE, YAKE, and KEYBERT, which are state-of-the-art methods for keyword extraction. We up-limit the number of keywords to 5 via hyperparameter search to mitigate possible noise. We observed that RAKE produced the most favorable results (see Table 5.13). Regarding the exact match over CONCEPTNET, if the exact match of k fails, we drop k from $\mathcal{K}$ (a scenario that occurs on average 1% of the time).

5.2.D Output Analysis

Figure 5.6 shows graphical examples of reasoning paths. Table 5.14 reports some erroneous predictions.

5.2.E Limitations and Future Directions

Future work should explore the following limitations.

Computational Intensity. Our solution requires the creation of a graph for each question, which is not demanding at experimental time (i.e., for training and testing in existing datasets) because graph creation is performed offline as a preprocessing phase. In contrast, in a real-world application characterized by a flow of new user questions, there will be a waiting time due to this procedure.

Knowledge Base Dependence. Our method requires a knowledge base to work, which could be challenging to find or create for a diverse range of cases, such as biomedical [Moro et al., 2022a, 2023e] and legal applications [Moro and Ragazzi, 2022, Moro et al., 2023a].

Reasoning Paths Evaluation. There are no automatic evaluation metrics that can capture the usefulness of the reasoning paths provided. Even human analysis is not as simple as it may seem because of the lack of a rigorous standard to follow.

Additional Tasks. As introduced in communication networks [Moro and Monti, 2012], tracking and propagating knowledge refinements between graph nodes can beneficial for creating interpretable pathways to support prediction in other real-world generative tasks, such as text summarization [Moro and Ragazzi, 2023, Moro et al., 2023c].

5.3
Enhancing Legal Question Answering with Data Generation and Knowledge Distillation from Large Language Models

Under review as: *Paolo Italiani, Luca Ragazzi, and Gianluca Moro. Enhancing legal question answering with data generation and knowledge distillation from large language models, 2024. Submitted to Artificial Intelligence and Law*

Abstract

Legal question answering (LQA) relies on supervised methods to automatically handle law-related queries. These solutions require a significant amount of carefully annotated data for training, which makes the process very costly. Although large language models (LLMs) show promise in zero-shot QA, their computational demands limit their practical use, making specialized small language models (SLMs) more favorable. Furthermore, the growing interest in synthetic data generation has recently surged, spurred by the impressive generation capabilities of LLMs. This paper presents ACE-ATTORNEY, an LLM distillation framework devised to develop LQA data and supervised models without human annotation. Given a textual prompt, a frozen instruction-tuned LLM generates artificial examples that are used as knowledge to train a student SLM with an order of magnitude fewer parameters. Extensive experiments prove the effectiveness and efficiency of distilled models on SYN-LEQA and PRIVACYQA, our synthetic dataset and a public human-annotated corpus, respectively. Notably, our best SLM achieves LLM-comparable performance on PRIVACYQA using just 30 training samples and ≈1200% less CO_2 emissions.

5.3.1 Introduction

The rapid introduction and growing availability of novel digitized legal texts result in an overload of information for legal professionals. Acting as an expert assistant, legal question answering (LQA) systems could help quickly capture the main points of new cases [Gil, 2023]. However, the need for accurate training data and its associated annotation cost represent a barrier for emerging applications. Consequently, there is a pressing demand for LQA data and models to streamline the deployment of production-ready solutions in high-value domains such as law [Louis et al., 2023].

Automated data generation has recently attracted enormous interest due to the remarkable generative capacity of large language models (LLMs) such as CHATGPT [OpenAI, 2023]. On the downside, the huge size of such models poses notable deployment obstacles (e.g., >190 GB of GPU RAM is needed to serve a 180B LLM with 8-bit quantization [Almazrouei et al., 2023]). Consequently, given such prohibitive costs, knowledge distillation (KD) [Hinton et al., 2015] has emerged as a strategy, creating task-specific

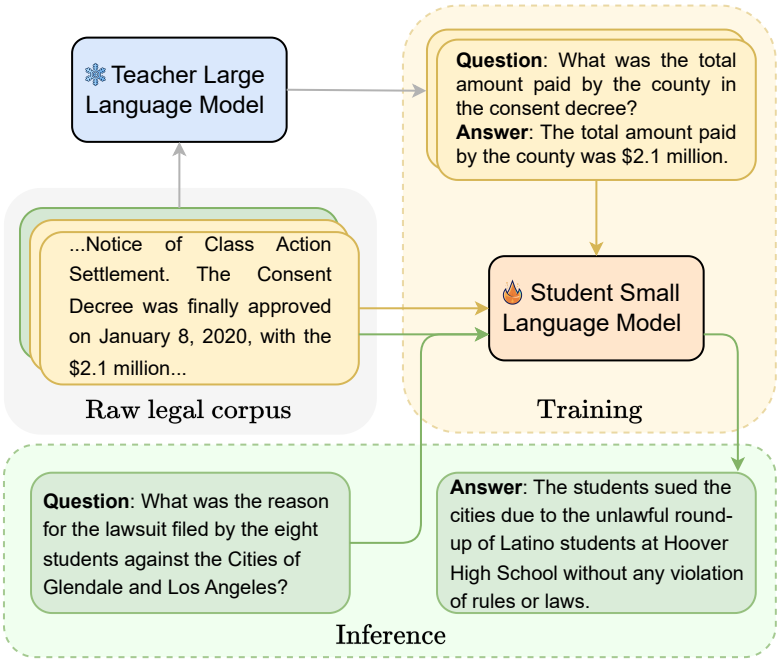


FIGURE 5.7. ACE-ATTORNEY overview. Along the yellow path, a student model is trained by leveraging the synthetic examples generated by a frozen LLM. The green path illustrates the inference phase, which entails answering a question based on a pertinent document.

small language models (SLMs) trained on synthetic data produced by LLMs [Meng et al., 2022a, Ye et al., 2022, Gao et al., 2023a]. This paradigm enables the creation of cost-efficient student models, affordable for most product teams, which often outperform prompt-based zero-shot LLMs despite having order-of-magnitude fewer parameters [Zhou et al., 2023a]. However, previous work has focused mainly on text classification, neglecting generative tasks such as QA. Furthermore, upon closer inspection, existing LQA datasets[11] exhibit at least one of the following limitations: (i) they concern specialized domains with constrained scope, such as tax law [Holzenberger et al., 2020]; (ii) they are restricted to multiple choice [Zheng et al., 2021, Bongard et al., 2022] or few-word replies [Sovrano et al., 2021]; (iii) the answer is a verbatim extraction from the context [Ravichander et al., 2019, Ahmad et al., 2020].

Motivated by the power of LLM text generation and the scarcity of LQA contributions, we present ACE-ATTORNEY (Figure 5.7), an LLM KD framework to automate the generation of LQA datasets and models. First, we use a frozen LLM to create a comprehensive synthetic dataset via carefully designed prompts on legal corpora, filling the dearth of publicly available LQA datasets. Subsequently, a student SLM is trained on the artificial question-answer pairs decoded by the teacher LLM, operating in a pure zero-shot setting without involving human annotations throughout the entire pipeline.

[11] As widely accepted by the community to track advancement progress, we analyzed contributions for English.

To fulfill the available hardware infrastructure, our framework can accommodate models of different sizes without presumption on their architecture.

We consider privacy policies and civil rights lawsuits as the raw legal knowledge to build SynLeQA, a new synthetic LQA dataset originated by LLM KD. We then fine-tune multiple million-scale SLMs on SynLeQA and perform in- and cross-domain experiments on a real human-annotated LQA dataset. Rigorous analysis shows that our distilled models, despite having order-of-magnitude fewer parameters, outperform neo-released state-of-the-art billion-scale LLMs with ≈1200% less CO_2 emissions. Furthermore, we show their ability to obtain LLM-comparable results with just a few dozen training samples.

5.3.2 Related Work

Legal Question Answering LQA has been a long-standing challenge for legal intelligence applications [Kano et al., 2018], fueled by the overwhelming influx of new judicial information. Consequently, to formulate and facilitate practical remedies, several datasets have been proposed. CaseHOLD [Zheng et al., 2021] collects more than 53K multiple-choice questions to identify the relevant holding of a cited case. PrivacyQA [Ravichander et al., 2019] aggregates 1750 questions on privacy policies of mobile applications. PolicyQA [Ahmad et al., 2020] provides 714 human-annotated inquiries for a wide range of privacy practices.

Synthetic Data Generation Several works focused on the fabrication of artificial data using generative models. Early attempts take advantage of human annotations to generate data samples and train models in a semi-supervised fashion [Anaby-Tavor et al., 2020, Kumar et al., 2020, Puri et al., 2020, Lee et al., 2021a]. As LLMs have become increasingly popular, they have also gained attention for generating task-specific examples [Yoo et al., 2021, Bonifacio et al., 2022, Carranza et al., 2023, Guo et al., 2023]. However, these solutions are still based on expert-made data. We advocate for a recent research pathway that explores a pure zero-shot scenario, fueled by LLM KD to fine-tune cost-efficient SLMs. Schick and Schütze [2021] use LLMs to generate labeled text pairs for the semantic textual similarity task. Meng et al. [2022a] produce samples using generation probability and regularization techniques for text classification. Yu et al. [2023] use complex attributed prompts, creating diverse and attributed artificial data. To investigate in more depth the zero-shot capabilities of LLMs and reduce their computational burden, Ye et al. [2022] train LSTMs with synthetic data.

5.3.3 Ace-Attorney

We introduce Ace-Attorney, a new framework devised to develop LQA-specific data and models through LLM KD.

Task Definition LQA can be formally defined as follows. Let $\mathbf{D} = \{\mathbf{d}_1, \ldots, \mathbf{d}_n\}$ be a corpus of legal documents. Given a single-sentence question $\mathbf{q}$, a retriever $\mathcal{R}$ is asked

to sort the documents in $\mathbf{D}$ by relevance: $\{\mathbf{d}'_1, \ldots, \mathbf{d}'_n\} = \mathcal{R}(\mathbf{D} \mid \mathbf{q})$. Then, a generative model $\mathcal{G}$ produces the answer $\mathbf{a}$ using both $\mathbf{q}$ and $\mathbf{d}'_1$, namely $\mathbf{a} = \mathcal{G}(\mathbf{q}, \mathbf{d}'_1)$.

5.3.3.1 Syn-LeQA

Motivated by the unavailability of LQA datasets, this work focuses on the generation of examples $\mathcal{E} = <\mathbf{q}, \mathbf{d}'_1, \mathbf{a}> \mid \mathbf{d}'_1 \in \mathbf{D}$, composing SYN-LEQA, our synthetic LQA dataset for training $\mathcal{G}$.

Challenges This data generation process faces several challenges. C1 The documents in $\mathbf{D}$ of the generated examples $\mathcal{E}_1, \ldots, \mathcal{E}_{|\mathcal{E}|}$ should not be overly similar to ensure generalizability across unseen data and domains, but should also not be too dissimilar to sidestep succumbing to a trivial retrieval task. C2 The questions must provide sufficient information for $\mathcal{R}$ to accurately retrieve the corresponding document in $\mathbf{D}$. C3 The answers should be coherent and factual w.r.t. the question and the source document, respectively.

Source Data For the fabrication of SYN-LEQA, we identify two diverse legal sources in $\mathbf{D}$. **(1)** *Civil rights lawsuits*: we consider the short summaries of the MULTI-LEXSUM dataset [Shen et al., 2022], a legal corpus for multi-document summarization, consisting of a description of the background, parties involved, and outcome of the case. **(2)** *Privacy policies*: we aggregate the policies provided by the training set of PRIVACYQA [Ravichander et al., 2019] and POLICYQA [Ahmad et al., 2020], which are contractual agreements that bind companies to their clients. To align with the input length of short summaries in MULTI-LEXSUM and increase the number of samples, we divided the original policies into chunks of a maximum of four sentences. Accordingly, by including instances from different legal domains, we also tackle C1.

LLM Annotation We use LLAMA-2-13B as LLM teacher $\mathcal{M}$.[12] Precisely, given a document $\mathbf{d}_i$, we provide $\mathcal{M}$ with a curated prompt template $\mathcal{T}(\cdot)$ to generate the question–answer pairs, formally defined as $(\mathbf{q}, \mathbf{a}) = \mathcal{M}(\mathcal{T}(\mathbf{d}_i))$. After careful prompt engineering (see Appendix 5.3.C), the following prompt emerges as our primary choice:

``Provide an extremely specific question and short answer (one sentence) based on a fact present in the following document: {$\mathbf{d}_i$}''

.

Quality Control In light of the above challenges, we first perform a careful post-processing to control the quality of the generated samples. (i) We remove instances with empty questions or answers. (ii) We address C2 by filtering samples with too generic questions using a retrieval-based approach. Technically, given $\mathbf{q}_i$, we exclude samples whose $\mathbf{d}_i$ is not retrieved in the first top-10 positions using BM25 OKAPI [Robertson

[12] Although ACE-ATTORNEY is model-agnostic, our hardware infrastructure is restricted to 24 GB of GPU RAM.

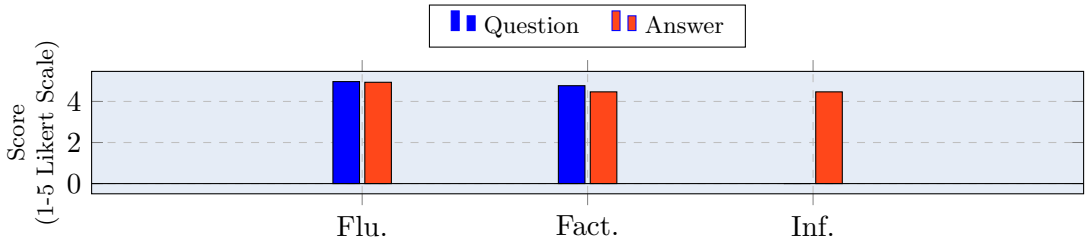


FIGURE 5.8. Human analysis on SYN-LEQA's data quality according to *fluency*, *factualness* and *informativeness*.

Algorithm 7 Selective Generative Paradigm

Input: $\mathbf{q}$ ▷ question
Output: $\mathbf{a}$ ▷ answer

$\{\mathbf{d}'_1, \ldots, \mathbf{d}'_n\} = \mathcal{R}(\mathbf{D} \mid \mathbf{q})$, $i = 1$
while $\mathbf{a} = \mathbf{s}$ **do**
  $\mathbf{a} = \mathcal{G}(\mathbf{q}, \mathbf{d}'_i)$
  $i = i + 1$
end while

et al., 1994] (more details are given in Section 5.3.4.4). Finally, to verify C3, we drive a human analysis to measure data quality. Specifically, we present 50 random samples to three English-proficient annotators asking to assign a score on a Likert scale from 1 (worst) to 5 (best), according to *fluency*, *factualness* (for both $\mathbf{q}_i$ and $\mathbf{a}_i$), and *informativeness* (for just $\mathbf{a}_i$). Figure 5.8 shows very high results, supported by inter-annotator agreement of 0.62 with Cohen's κ coefficient, revealing the great generative capability of LLMs. Details on the evaluation criteria are provided in Appendix 5.3.B.

5.3.3.2 Selective Generative Paradigm

The quality of the generated answer is highly dependent on $\mathcal{R}$, which prevents $\mathcal{G}$ from producing a coherent and factual response when retrieving the wrong document. Consequently, previous work has put great effort into improving communication between $\mathcal{R}$ and $\mathcal{G}$, devising solutions that combine these two components in an end-to-end fashion [Lewis et al., 2020b, Lai et al., 2023]. However, $\mathcal{G}$ still places complete trust in $\mathcal{R}$, without having the opportunity to replace the received document.

We advocate for a $\mathcal{G}$ capable of doubting the conduct of $\mathcal{R}$. We term this ability as *selective generative paradigm* (SGP), which allows $\mathcal{G}$ decide whether to generate the answer based on the current document retrieved or ask $\mathcal{R}$ to return a different one. Technically, if $\mathcal{G}$ is not satisfied with the selection, it must produce the following sentence $\mathbf{s}$: `''Please provide the correct document.''` Therefore, the second most appropriate document, according to $\mathcal{R}$, is passed to $\mathcal{G}$, which re-assesses the relevance until it generates a proper answer. The whole procedure is depicted in Algorithm 7.

TABLE 5.15. Dataset statistics. The word count is calculated with the NLTK tokenizer [Wagner, 2010].

Set	# Samples	# Question Words	# Answer Words	# Doc Words
Syn-LeQA				
Train	3854	27.55	34.50	115.90
Dev	481	27.66	34.46	118.39
Test	483	27.82	34.32	116.31
PrivacyQA				
Test	431	9.37	42.48	116.70

5.3.4 Experimental Setup

5.3.4.1 Dataset

In addition to the in-domain experiment on the proposed SYN-LEQA, we perform a cross-domain evaluation on the PRIVACYQA dataset [Ravichander et al., 2019], aiming to evaluate the distilled SLMs trained on SYN-LEQA. Since this benchmark is centered on a classification task, we make adjustments to align it with our specific generative QA objective. In PRIVACYQA, for each query $\mathbf{q}_i$, expert annotators classified each sentence in the corresponding document $\mathbf{d}_i$ as relevant or not. Therefore, considering that there can be multiple sentences marked as relevant for $\mathbf{q}_i$, we select the longest as the candidate target answer $\mathbf{a}_i$. Table 5.15 provides the statistics of the datasets.

5.3.4.2 Metrics

To comprehensively assess model performance, we cover multiple evaluation dimensions. First, we use recall-oriented **ROUGE-{1,2,L}** [Lin, 2004] and precision-focused **BLEU-{1,2,3,4}** [Papineni et al., 2002]. Inspired by Moro et al. [2023b], we report on their aggregated score, $\mathcal{RG}$ and $\mathcal{B}$, respectively. Moving beyond lexical superficiality, we employ **BERTScore** (BeS) [Zhang et al., 2020b] and **BARTScore** (BaS) [Yuan et al., 2021] for semantic coverage, two model-based metrics that exhibit strong correlations with human judgment. Furthermore, to compare the efficiency of models, we monitor the **inference runtime** and compute **CO_2 emissions** with CodeCarbon.[13] Finally, we condense our judgment to weigh both cost and effectiveness in a single score using **Carburacy** [Moro et al., 2023b]. The technical details are given in Appendix 5.3.A.2.

5.3.4.3 Models

SLMs We consider **Bart** [Lewis et al., 2020a], one of the most popular models for generative tasks characterized by a denoising pretraining objective, and **Flan-T5** [Chung et al., 2022], a model fine-tuned with instructions on a mixture of text-to-text tasks. We benchmark both models with different sizes, ranging from 60M to 780M parameters.

[13] https://github.com/mlco2/codecarbon

TABLE 5.16. Retrieval results. Bold and underline denote the best and second-best scores.

Model	R@1	R@5	R@10	Run.	Emis.
BM25 OKAPI	83.02	95.65	97.10	**11.80**	1.20
BiE DISTILROBERTA	70.81	87.78	90.89	33.56	<u>0.69</u>
BiE MPNET	73.50	87.16	91.72	<u>29.18</u>	**0.63**
CE TINYBERT	<u>89.65</u>	<u>96.80</u>	<u>98.14</u>	1473.71	32.15
CE MINILM	**90.06**	**97.52**	**98.76**	1428.88	162.58

TABLE 5.17. Quantitative results on SYN-LEQA's test set. Bold and underline denote the best and second-best results.

Syn-LeQA													
			Syntactic					Semantic		Efficiency			
Model	Size	Release	$\mathcal{B}$	R-1	R-2	R-L	$\mathcal{RG}$	BaS-F1	BeS	Carb.	Run.	Emis.	R@1
Fine-tuned SLMs													
FLAN-T5-small	60M	2022/10	44.09	56.31	44.00	56.02	51.94	-2.15	55.86	75.14	195.49	6.66	75.78
BART-base	149M	2020/07	46.80	62.66	50.31	61.96	58.12	-1.92	62.21	78.88	161.17	7.42	88.61
FLAN-T5-base	250M	2022/10	49.70	63.48	50.68	62.96	58.83	-1.84	64.33	79.28	240.16	8.43	84.47
BART-large	406M	2020/07	<u>52.51</u>	<u>65.85</u>	<u>53.77</u>	<u>65.34</u>	<u>61.46</u>	<u>-1.79</u>	<u>66.06</u>	<u>80.78</u>	249.15	9.86	<u>89.03</u>
FLAN-T5-large	780M	2022/10	**53.37**	**66.96**	**54.70**	**65.88**	**62.32**	**-1.70**	**67.59**	**81.07**	481.16	21.61	**90.48**
Zero-shot LLMs													
LLAMA-2	7B	2023/07	36.38	50.39	37.95	49.15	45.69	-2.78	41.72	69.74	2390.76	98.79	82.61
MISTRAL	7B	2023/09	12.76	41.94	29.56	41.17	37.44	-2.83	36.79	59.01	6650.08	512.61	54.04
ZEPHYR	7B	2023/10	16.45	57.56	43.00	54.49	51.48	-1.92	50.84	63.36	7070.25	883.53	82.19
ORCA-2	7B	2023/11	22.41	50.05	36.32	48.29	44.72	-2.66	36.09	64.98	5814.93	410.86	81.37
NEURAL-CHAT	7B	2023/11	25.18	54.11	34.76	49.77	45.90	-2.64	45.67	68.15	4346.67	223.94	82.19
STARLING	7B	2023/11	33.13	58.69	45.08	56.57	53.26	-2.43	54.32	73.45	4687.27	172.07	82.19
NOTUS	7B	2023/11	13.16	50.03	36.33	48.21	44.69	-2.12	46.08	42.20	30687.14	3237.09	77.23
LLAMA-2	13B	2023/07	35.59	45.77	33.52	44.05	40.99	-2.99	35.41	64.86	3871.89	225.12	80.33
ORCA-2	13B	2023/11	12.21	43.04	30.12	41.84	38.20	-2.49	33.85	59.36	14176.05	527.61	77.85

For the fine-tuning process, we concatenate the question and the relevant document and train them to produce the corresponding answer.

LLMs We examine **LlaMa-2-{7B,13B}** [Touvron et al., 2023b], which is an upgraded version of LLAMA-1 [Touvron et al., 2023a] pretrained on additional 40% of data. **Mistral-7B** [Jiang et al., 2023] comprises grouped-query attention [Ainslie et al., 2023] and sliding window attention [Beltagy et al., 2020] to achieve a harmonious equilibrium between the pursuit of high performance and efficiency. **Zephyr-7B** [Tunstall et al., 2023] leverages MISTRAL-7B as a starting checkpoint, distilling conversational capabilities with direct preference optimization from a dataset of AI feedback. **Orca-2-{7B,13B}** [Mitra et al., 2023] is a refined iteration of LLAMA-2 that outperforms models of comparable scale on tasks requiring advanced reasoning abilities. **Neural-chat-7B** is a model based on MISTRAL-7B fine-tuned on SLIMORCA. **Starling-7B** is trained with reinforcement learning from AI harnessing NECTAR, a GPT-4 labeled ranking dataset composed of chat prompts. Lastly, through fine-tuning on a refined variant of ULTRAFEEDBACK [Cui et al., 2023], **Notus-7B** [Bartolome et al., 2023] elevates the performance of ZEPHYR-7B.

5.3.4.4 Document Retrieval

The retriever is a pivotal component in real-world QA pipelines, having a direct influence on the overall accuracy, efficiency, and reliability of the entire solution. In light of this, we perform a comparison of different methods with the aim of maximizing performance while minimizing costs. First, we test **BM25 Okapi** [Robertson et al., 1994], a probabilistic model that estimates the relevancy of documents w.r.t. a query as follows:

$$\text{BM25} = \sum_{t \in q} \log \left(\frac{n - f_t + 0.5}{f_t + 0.5} \right) \cdot \text{TF}_{\text{BM25}} \tag{5.6}$$

$$\text{TF} = \frac{f_{t,\mathbf{d}} \cdot (k_1 + 1)}{f_{t,\mathbf{d}} + k_1 \cdot ((1 - b) + (b \cdot \ell_{\mathbf{d}} / \ell_{\text{avg}}))} \tag{5.7}$$

where n is the number of documents in the corpus, f_t is the number of documents containing the query term t, $f_{t,\mathbf{d}}$ is the number of occurrences of t in the document $\mathbf{d}$, $\ell_{\mathbf{d}}$ is the number of terms in $\mathbf{d}$, and ℓ_{avg} is the average of $\ell_{\mathbf{d}}$ over the corpus. We use the standard parameters: $k_1 = 1.2$ and $b = 0.75$.

Furthermore, we evaluate the *bi-encoder* (BiE) architecture [Reimers and Gurevych, 2019], which uses a siamese BERT network to create semantically meaningful sentence embeddings. This involves independent processing of $\mathbf{q}$ and $\mathbf{d}_i$ through an encoder-only transformer, resulting in two unique sentence embeddings that can be compared using cosine similarity. To enhance computational efficiency, we conduct indexing using FAISS [Johnson et al., 2021], which tackles the problem of efficient comparison between high-dimensional vectors. Lastly, we assess the *cross-encoder* (CE) implementation [Chen et al., 2020a], using an encoder-only transformer to process the concatenated $\mathbf{q}$ and $\mathbf{d}_i$, producing a score indicating the pertinence of the document. Looking at the MS Marco Passage Reranking [Nguyen et al., 2016] and TREC [Craswell et al., 2020] benchmarks, we operate **MPNet** [Song et al., 2020] and **DistilRoBERTa** [Sanh et al., 2019] as bi-encoders, and **TinyBERT** [Jiao et al., 2020] and **MiniLM** [Wang et al., 2020c] as cross-encoders.

Results Table 5.16 shows the retrieval scores on the test set of Syn-LeQA. Technically, we report the Rank@k metric, with $k \in \{1, 5, 10\}$, assessing how frequently the correct document is found within the first k retrieved items. Furthermore, we provide the runtime in seconds and the CO_2 emissions in milligrams. In line with Reimers and Gurevych [2019], cross-encoders exhibit the best results at the price of significantly higher computational costs because they require computing a score for every possible combination of query and document. For example, although CE MiniLM demonstrates superior performance, it requires over ten times more time and emissions than BM25 Okapi. However, this performance gap decreases as k increases. For this reason, we select BM25 Okapi as retrieval for our experiments, aiming to strike the best equilibrium between performance and computational overhead.

TABLE 5.18. Classification results of SGP. Bold and underline denote the best and second-best results.

Model	Size	Precision	Recall	F1	Accuracy
Fine-tuned SLMs					
FLAN-T5-small	60M	56.63	57.32	56.97	85.30
BART-base	149M	<u>86.30</u>	<u>76.83</u>	81.29	<u>94.00</u>
FLAN-T5-base	250M	79.17	69.51	74.03	91.72
BART-large	406M	84.72	74.39	**96.06**	93.37
FLAN-T5-large	780M	**86.67**	**79.27**	<u>82.80</u>	**94.41**
Zero-shot LLMs					
LLAMA-2	7B	0.00	0.00	0.00	82.40
MISTRAL	7B	20.79	51.22	29.58	58.59
ZEPHYR	7B	68.42	15.85	25.74	84.44
ORCA-2	7B	61.90	15.85	25.24	84.06
NEURAL-CHAT	7B	85.19	28.05	42.20	86.96
STARLING	7B	78.95	18.29	29.70	85.30
NOTUS	7B	48.15	31.71	38.24	82.61
LLAMA-2	13B	27.03	12.20	16.81	79.50
ORCA-2	13B	53.70	35.37	42.65	83.85

5.3.4.5 Hardware Setup and Environment

All runs are tracked with Weights & Biases[14] and executed on a workstation with a single Nvidia GeForce RTX3090 GPU of 24 GB of dedicated memory, 64 GB of VRAM, and an Intel® Core™ i9-10900X1080 CPU @ 3.70GHz. The reference operating system is Ubuntu 20.04.3 LTS. To enhance consistency and portability, our development environment is built on top of a Docker container with a NVIDIA image.[15] Experimental details are given in Appendix 5.3.A.

5.3.5 Results and Discussion

Effectiveness Table 5.17 presents the results on SYN-LEQA using SGP. Across both syntactic and semantic metrics, distilled SLMs consistently outperform LLMs. To be precise, the biggest improvement is recorded compared to LLAMA-2-13B, ORCA-2-13B and MISTRAL-7B. The performance gap is reduced compared to the newly introduced STARLING-7B, ZEPHYR-7B and NEURAL-CHAT-7B. Of particular interest are the enhanced retrieval results obtained thanks to SGP. Specifically, SLMs excel at identifying improperly retrieved documents, whereas this method appears to hinder retrieval performance for LLMs. The ability of the models to classify the retrieved documents as relevant or not is summarized in Table 5.18. Technically, we label the generated answer as 1 when the model outputs **s** (indicating an incorrectly retrieved document) and 0 otherwise. These findings show the benefits of SGP, which significantly increases on

[14] https://wandb.ai
[15] nvidia/cuda:11.3.1-devel-ubuntu20.04

PrivacyQA												
			Syntactic					Semantic		Efficiency		
Model	**Size**	**Release**	$\mathcal{B}$	**R-1**	**R-2**	**R-L**	$\mathcal{RG}$	**BaS-F1**	**BeS**	**Carb.**	**Run.**	**Emis.**
Fine-tuned SLMs												
FLAN-T5-small	60M	2022/10	17.21	30.65	19.68	32.01	27.36	-3.34	30.54	56.86	102.12	7.54
BART-base	139M	2020/07	11.40	28.31	16.29	29.85	24.73	-3.45	28.71	54.44	80.47	5.70
FLAN-T5-base	250M	2022/10	13.12	29.68	19.13	31.31	26.63	-3.32	29.92	56.19	127.07	7.83
BART-large	406M	2020/07	18.03	33.04	20.70	34.10	29.17	-3.26	32.92	58.40	160.82	11.74
FLAN-T5-large	780M	2022/10	18.00	35.16	**23.20**	35.90	**31.31**	-3.15	34.55	**60.19**	284.08	14.18
Zero-shot LLMs												
LLAMA-2	7B	2023/07	9.18	27.94	12.45	27.56	22.53	-3.62	27.95	50.80	1693.14	131.46
MISTRAL	7B	2023/09	12.90	29.08	14.31	29.21	24.08	-3.52	28.39	51.06	2497.99	238.00
ZEPHYR	7B	2023/10	11.57	37.65	19.43	35.34	30.61	**-2.85**	31.29	52.52	3683.45	597.87
ORCA-2	7B	2023/11	11.07	26.39	8.78	24.34	19.71	-4.03	14.85	47.21	2632.77	199.24
NEURAL-CHAT	7B	2023/11	5.43	24.32	6.02	22.66	17.55	-4.09	21.24	44.32	2282.09	257.43
STARLING	7B	2023/11	**19.83**	**38.22**	19.64	**36.30**	31.17	-3.63	**34.66**	57.81	3220.92	183.61
NOTUS	7B	2023/11	11.12	36.99	16.70	33.06	28.70	-3.01	32.38	41.81	11047.64	1684.49
LLAMA-2	13B	2023/07	12.07	27.65	12.75	27.59	22.55	-3.56	27.54	50.55	2823.87	155.35
ORCA-2	13B	2023/11	7.28	25.61	9.89	24.26	19.82	-3.40	19.93	41.83	13044.20	791.71

TABLE 5.19. Quantitative results on PRIVACYQA's test set. Bold and underline denote the best and second-best results.

average the original R@1 score of 83.02, as shown in Table 5.16. In addition, Table 5.19 shows the results on a real human-annotated dataset, where we test the SLMs trained on SYN-LEQA. Again, SLMs overall outperform LLMs, although they have a much smaller number of parameters. Figure 5.9 visually summarizes the results.

Efficiency In addition to the better effectiveness of distilled SLMs, from Table 5.17 and Table 5.19, we can also appreciate the radical reduction in carbon emissions and runtime at inference time. For example, the best-performing LLM on PRIVACYQA, STARLING-7B, is characterized by ≈1200% increment in CO_2 emissions compared to

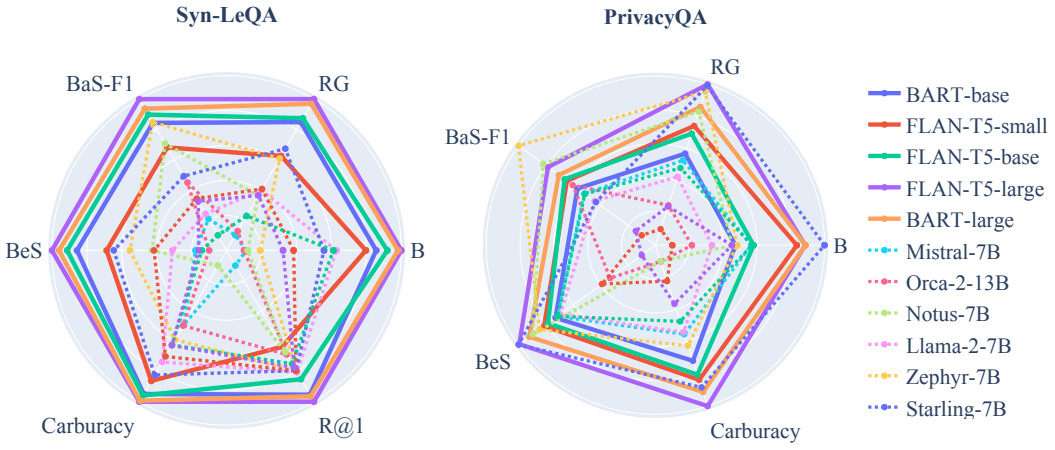


FIGURE 5.9. Graphical overview of model effectiveness.

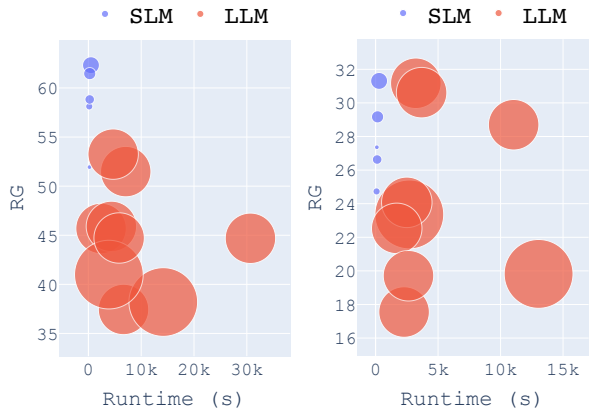


FIGURE 5.10. Performance–efficiency relation (left SYN-LEQA, right PRIVACYQA). The size of bubbles refers to the number of model parameters.

the best SLM, FLAN-T5-large. Interestingly, Figure 5.10 clearly highlights that SMLs (blue bubbles) achieve higher $\mathcal{RG}$ scores with a significantly shorter runtime.

Few-shot Setting It is plausible that a real-world organization lacks adequate hardware resources to generate thousands of synthetic examples. Additionally, there might be a need for a thorough examination of the created question–answer pairs before integrating them as references for the production model. Accordingly, it is of particular interest to study the performance of the student model w.r.t. the number of synthetic training instances available. Figure 5.11 summarizes the results of FLAN-T5-LARGE fine-tuned on an increasing number of SYN-LEQA samples, drawn equally from both lawsuits and policies. Notably, the SLM surpasses STARLING-7B on SYN-LEQA with only 20 labeled examples, and achieves comparable performance on PRIVACYQA using only 30 samples.

Prompt Engineering As emphasized in previous studies [Khashabi et al., 2022, Wei et al., 2022b, Ye et al., 2022], the configuration of the prompt template can significantly influence the process of synthetic data generation. In this segment, we discuss its impact on fine-tuning FLAN-T5-large when applied to PRIVACYQA. Specifically, we

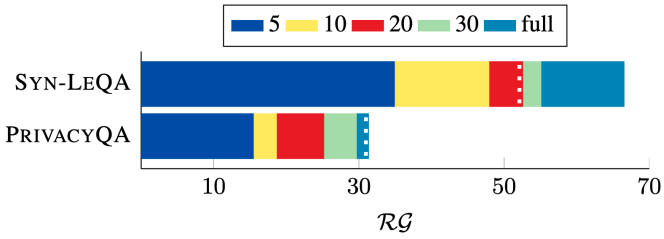
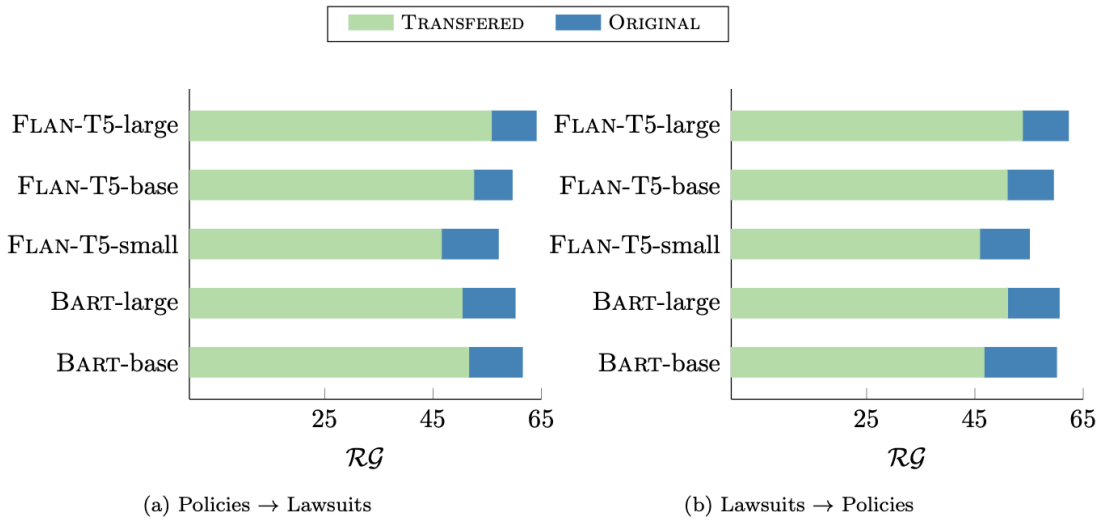


FIGURE 5.11. Few-shot setting results. The dashed white bands mark the top-performing LLM.



(a) Policies → Lawsuits (b) Lawsuits → Policies

FIGURE 5.12. Transferability results on SYN-LEQA. TRANSFERED denotes a model trained on one dataset and tested on the other. ORIGINAL denotes a model trained and tested on the same dataset.

analyze four different prompting strategies: (i) *zero-shot task description*, we provide a textual description of the task and a test example; (ii) *few-shot prompting*, the model is given a few demonstrations of the task at inference time, but no weight updates are allowed; (iii) *zero-shot* LLAMA *task description*, we include the same special tokens used during LLAMA-2 training with a wide variety of system prompts intended for different tasks; (iv) *few-shot with task description*. In an effort to reduce computational expenses, we conduct this analysis using 100 instances drawn equally from lawsuits and policies. Table 5.20 indicates that the *zero-shot task description* performs the best, while the *zero-shot* LLAMA *task description* exhibits slightly higher efficiency, albeit with reduced effectiveness compared to the former. As expected, strategies that utilize few-shot prompting take longer to generate question-answer pairs because the model is provided with longer instructions as input. Moreover, this appears to harm the data generation process, resulting in significantly inferior results compared to methods that abstain from few-shot prompting.

Transfer Learning We evaluate the transferability of SLMs by assessing their performance when trained on one subset of SYN-LEQA (i.e., lawsuits or policies) and tested on the other. We report the results with the same subset for training and test as upper bound. Figure 5.12 demonstrates the adaptability of SLMs, showcasing consistent performance when transferred between different domains.

5.3.6 Conclusions

In this paper, we introduce ACE-ATTORNEY, a new legal framework designed to produce LQA-specific datasets and supervised SLMs through LLM KD. Precisely, given an

TABLE 5.20. Quantitative results on PRIVACYQA's test set using different prompt templates. Bold and underline denote the best and second-best results.

Prompt	$\mathcal{RG}$	BeS	Run.
`Few-shot prompting`	21.47	24.74	10672
`Task description`	**30.88**	**33.92**	<u>1591</u>
`LLAMA task description`	<u>25.48</u>	<u>28.44</u>	**1462**
`Few-shot prompt. w/ task descr.`	22.86	25.87	5061

input textual prompt, a state-of-the-art frozen LLM generates artificial samples that are used as knowledge to train a reduced-parameter student model. We collect LLM-crafted samples and create SYN-LEQA, our synthetic dataset proposed to address the scarcity of public LQA corpora. Experiments carried out on both SYN-LEQA and a real expert-labeled dataset, PRIVACYQA, showcase the ability of student models to surpass the latest cutting-edge LLMs. Importantly, distilled models operate with significantly fewer parameters and demand significantly shorter inference time and CO_2 emissions. Moreover, our emphasis on the few-shot scenario guarantees competitive performance even when we have access to a limited number of synthetic instances. This approach allows manual oversight at a reasonable cost across the entire generated dataset.

5.3.7 Impact Statement and Limitations

Our work has great potential to promote new LQA datasets and models. However, due to the well-known hallucination problems of LLMs, experts should verify the quality of the generated data for real-world usage. Despite the large amount of up-to-date LLMs employed in our experiments, we focused on models capable of operating within the constraints of a single 24 GB of GPU RAM. Consequently, this excluded larger and potentially higher-performing solutions from our study, also affecting the synthetic dataset generation process, which was carried out exclusively using LLAMA-2-13B. Furthermore, our study's scope regarding the evaluation of our fine-tuned SLMs on real human-annotated datasets is constrained, as we considered only a single reference dataset in our analysis. However, this decision is due to PRIVACYQA being the only dataset that meets the necessary criteria required to act as a benchmark for generative LQA.

5.3.A Reproducibility

5.3.A.1 Implementation and Training Details

Models Table 5.21 enumerates the models used in this study, specifying their corresponding HuggingFace checkpoint.

Hyperparameters We list the hyperparameters used for SLMs fine-tuning and inference, and zero-shot LLMs.

TABLE 5.21. List of model checkpoints used in this study.

Model	Checkpoint
SLMs	
BART-base	facebook/bart-base
BART-large	facebook/bart-large
FLAN-T5-small	google/flan-t5-small
FLAN-T5-base	google/flan-t5-base
FLAN-T5-large	google/flan-t5-large
LLMs	
LLAMA-2-7B	meta-llama/Llama-2-7b-chat-hf
LLAMA-2-13B	meta-llama/Llama-2-13b-chat-hf
MISTRAL-7B	mistralai/Mistral-7B-Instruct-v0.1
ZEPHYR-7B	HuggingFaceH4/zephyr-7b-alpha
ORCA-2-7B	microsoft/Orca-2-7b
ORCA-2-13B	microsoft/Orca-2-13b
NEURAL-CHAT-7B	Intel/neural-chat-7b-v3-1
STARLING-7B	berkeley-nest/Starling-LM-7B-alpha
NOTUS-7B	argilla/notus-7b-v1

TABLE 5.22. Hyperparameters utilized for SLMs fine-tuning and inference, and zero-shot LLMs. † and ‡ are specific for SLMs and LLMs, respectively.

Hyperparameter	
Dropout rate	0.1 †
Learning rate	5e-5, linear scheduler †
Optimizer	0.9 β_1, 0.999 β_2, 1e-2 weight decay †
Batch size	2 †
Epochs	5 (validation every epoch) †
Decoding strategy	greedy search, max_length=300
Seed	42
Quantization	8-bit ‡

5.3.A.2 Metrics

Table 5.23 lists the hyperparameters of the metrics.

To increase interpretability and avoid slight range variations, we calculate BERTScore with `rescale_with_baseline=True`. Note that BARTScore computes the generation probability $p(\mathbf{y}|\mathbf{x}, \theta)$ of a sequence $\mathbf{y}$ conditioned on another sequence $\mathbf{x}$, where θ are the weights of a BART model. Due to this generative approach, the evaluation dimensions vary depending on how $\mathbf{y}$ and $\mathbf{x}$ are defined. In particular, we consider the Recall, Precision and F1 settings. Recall ($\mathbf{h} \rightarrow \mathbf{r}$, $p(\mathbf{r}|\mathbf{h}, \theta)$) quantifies how easily a gold reference ($\mathbf{r}$) could be generated by hypothesis ($\mathbf{h}$). Precision ($\mathbf{r} \rightarrow \mathbf{h}$, $p(\mathbf{h}|\mathbf{r}, \theta)$) evaluates how likely the answer hypothesis could be constructed based on the gold ref-

Metric	Bound	Hyperparameters
BLEU	$[0, 1]$	`bleu_types=["bleu1", "bleu2", "bleu3", "bleu4"]`, `weights="equidistributed"`
ROUGE	$[0, 1]$	`rouge_types=["rouge1","rouge2","rougeL"]`, `use_aggregator=True`, `use_stemmer=True`, `metric_to_select="fmeasure"`
BERTScore	$[-1, 1]$*	`model_type="bert-base-uncased"`, `idf=True`, `batch_size=64`, `nthreads=4`, `rescale_with_baseline=True`,* `use_fast_tokenizer=False`, `return_average_scores=False`
BARTScore	$]-\infty, 0[$	`model_checkpoint="facebook/bart-large-cnn"`, `batch_size=4`, `segment_scores=False`

TABLE 5.23. Hyperparameters initialization for utilized NLG metrics.

TABLE 5.24. Explanations on human evaluation aspect scales.

	Informativeness:
1	Answer is not relevant to the question
2	Answer is partially relevant and misses the main point of the question
3	Answer is relevant, but misses the main point of the question
4	Answer successfully captures the main point of the question but some relevant content is missing
5	Answer successfully captures the main point of the question
	Factualness:
1	Answer/question consists almost entirely of fabricated content that does not occur in the source document
2	Answer is mainly composed of hallucinations
3	Answer/question contains few hallucinations, but concern significant aspects of the original document
4	Answer contains few hallucinations, but are restricted to negligible facts
5	Answer/question is faithful with respect to the original document
	Fluency:
1	Answer/question is full of garbage fragments and is hard to understand
2	Answer/question contains fragments, missing components but has some fluent segments
3	Answer/question contains some grammar errors but is in general fluent
4	Answer/question has relatively minor grammatical errors
5	Fluent answer/question

erence. F score ($\mathbf{h} \leftrightarrow \mathbf{r}$) takes the arithmetic average of recall and precision.

Finally, inspired by Moro et al. [2023b], we also measure an aggregated ROUGE judgment: $\mathcal{RG} = {}^{\text{avg}(r_1,r_2,r_L)}/_{1+\sigma_r^2}$, where σ_r^2 is the F1 variance.[16] We apply the same principle to the BLEU-1/2/3/4 scores obtaining $\mathcal{B} = {}^{\text{avg}(b_1,b_2,b_3,b_4)}/_{1+\sigma_b^2}$.

Fine-tuning SLMs Our code is based on Python 3.8.10, PyTorch 1.12.1+cu113 [Paszke et al., 2019], and HuggingFace Transformers [Wolf et al., 2020]. We train each

[16] $\mathcal{RG}$ penalizes model results with discrepant unigram, bigram, and longest common subsequence overlaps.

TABLE 5.25. Qualitative examples on SYN-LEQA.

Document
In August 2006, the EEOC brought this suit in the U.S. District Court for the District of New Jersey on behalf of two employees over aged 60 who received reduced severance pay after their plant closed. It alleged that reducing the severance pay of employees over 60 who qualified for pensions violated the Age Discrimination in Employment Act. On May 27, 2008, the court ruled in favor of the defendants and awarded no relief.
QUESTION
Did the court rule in favor of the EEOC in the case of reduced severance pay for employees over 60 who qualified for pensions?
ANSWER
No, the court ruled in favor of the defendants and awarded no relief.
Document
We may suspend or terminate your rights to use the Parsec Properties (including your Account) at any time for any reason at our sole discretion, including for any use of the Parsec Properties in violation of these Terms. Upon termination of your rights under these Terms, your Account and right to access and use the Parsec Properties will terminate immediately. In addition, we use "pixel tags" (also referred to as clear GIFs, Web beacons, or Web bugs). Pixel tags are tiny graphic images with a unique identifier, similar in function to cookies, which are used to track online movements of Web users.
QUESTION
What is one specific reason that Parsec Properties may suspend or terminate a user's account and rights to use the Parsec Properties?
ANSWER
One specific reason that Parsec Properties may suspend or terminate a user's account and rights to use the Parsec Properties is for any use of the Parsec Properties in violation of these Terms.

model for five epochs and select the best checkpoints on the validation set with the ROUGE-1 score. We choose the AdamW optimizer [Loshchilov and Hutter, 2019].

Prompt-Based LLMs To make inference for zero-shot LLMs possible with our hardware, we exploit PEFT.[17] Precisely, we carry out 8-bit model quantization, enclosing the provided prompt within the recommended template for every model.

5.3.B Quality Control

To avoid subjectivity, Table 6.58 reports instructions provided to human annotators. Finally, Table 5.25 shows examples of SYN-LEQA.

5.3.C Prompt Engineering Details

Table 5.26 presents all the prompts used in our experiments.

[17] https://github.com/huggingface/peft

TABLE 5.26. Prompts engineered for synthetic dataset generation and zero-shot LLMs. † denotes the prompt chosen for the creation of SYN-LEQA.

Setting	Prompt
	Synthetic Dataset Generation
`Task description` †	Provide an extremely specific question and short answer (one sentence) based on a fact present on the following document: $\{d_i\}$. It should have the following format: Question: generated_question Answer: generated_answer.
`LLAMA task description`	$< s >$[INST] <<SYS>> You are an annotator that needs to provide an extremely specific question and short answer (one sentence) based on a fact present on a document. It should have the following format: Question: generated_question Answer: generated_answer. <</SYS>> The document is$\{d_i\}$ [/INST]
`Few-shot prompting`	Relevant document: $\{relevant_document_1\}$ Corresponding question and answer: Question: $\{question_1\}$ Answer: $\{answer_1\}$ Relevant document: $\{relevant_document_2\}$ Corresponding question and answer: Question: $\{question_2\}$ Answer: $\{answer_2\}$ Relevant document: $\{d_i\}$ Corresponding question and answer:
`Few-shot prompting w/ task description`	Provide an extremely specific question and short answer (one sentence) based on a fact present on the following document: $\{d_i\}$. It should have the following format: Question: generated_question Answer: generated_answer. Examples of questions and answers: Question: $\{question_1\}$ Question: $\{question_2\}$ Corresponding question and answer:
	Zero-shot LLM
`Prompting`	Provide a single sentence answer to the following question: $\{q_i\}$. Using the following document: $\{d_i\}$
`Prompting w/ selective generative paradigm`	Provide a single sentence answer to the following question: $\{q_i\}$. Using the following document: $\{d_i\}$ If the question is unanswerable given the provided document, write "Please provide the correct document."

5.4
Un*kn*own Claims: Generation of Fact-Checking Training Examples from Unstructured and Structured Data

Under review as: *Jean Flavien Bussotti, Luca Ragazzi, Giacomo Frisoni, Gianluca Moro, and Paolo Papotti. Unknown claims: Generation of fact-checking training examples from unstructured and structured data, 2024. Submitted to the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024*

Abstract

Computational fact-checking relies on supervised models to verify whether a claim is supported or refuted based on evidence provided. A substantial volume of meticulously annotated data is required for training, implying a resource-intensive and time-consuming annotation endeavor. We introduce UNOWN, a novel framework designed to automatically generate training instances for fact-checking systems from heterogeneous knowledge sources. Specifically, UNOWN ingests tabular and/or textual content, constructs evidence documents, and builds artificial claims that support or refute evidence. UNOWN is conceived to be efficient and module-centric, accommodating various evidence selection and claim generation strategies. Quantitative and qualitative experiments on FEVEROUS and SCIFACT, two benchmarks from different domains, reveal that our synthetic examples exhibit a quality similar to that of expert-labeled data.[18]

5.4.1 Introduction

In the realm of natural language inference (NLI), computational fact-checking involves determining the veracity of a written claim given a set of data as evidence. Evidence can consist of structured tabular information, unstructured text, or both, adding marked complexity to the task [Guo et al., 2022b].

Typically, experts carefully annotate and curate high-quality claim–evidence pairs to train fact-checking models. However, this methodology requires knowledge and incurs high costs [Nakov et al., 2021]. Moreover, in the present, information can rapidly proliferate as emerging domains take shape. As a case in point, consider the need to combat misinformation on social media and verify statements about COVID-19 [Saakyan et al., 2021, Carey et al., 2022]. Developing a production-ready model depends on accurate training data to effectively address new and sensitive demands. At the same time, annotation costs act as a barrier to meeting such urgent application requirements.

Previous work has introduced different methods with pretrained language models (PLMs) to forge labeled instances for fact-checking systems [Pan et al., 2021, Wright

[18]The code, data, and models are available at `https://anonymous.4open.science/r/unown-01B2`.

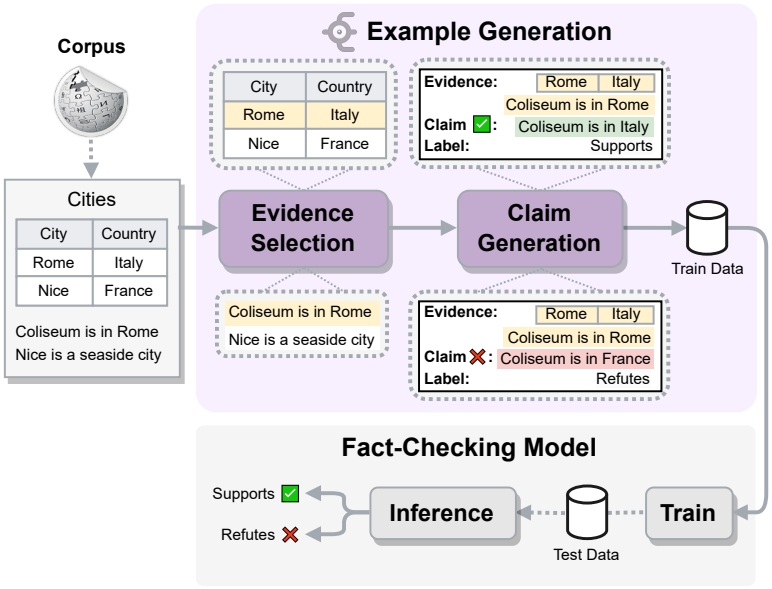


FIGURE 5.13. UNOWN fact-checking pipeline. Given a corpus of documents, the *Example Generation* module (investigated in this work) outputs supporting and refuting training instances.

et al., 2022]. However, these solutions have limitations: (i) they are restricted to certain fields due to the reliance on domain-specific knowledge bases (KBs); (ii) they do not accommodate the combination of structured and unstructured data as evidence, although it may be crucial for claim validation [Chen et al., 2020b, Aly et al., 2021]. See Table 5.27 for more details.

To fill this gap, we present UNOWN,[19] a new framework to automatically generate synthetic examples for fact-checking systems using multimodal evidence sources (Figure 5.13). UNOWN is modular and scalable, accommodating various strategies for evidence selection and claim generation, effortlessly adapting to different real-world scenarios. In fact, UNOWN works for any domain lacking external KB to produce refuting claims.

We conduct extensive experiments on two publicly available datasets that span general-domain and scientific contexts, namely FEVEROUS [Aly et al., 2021] and SCIFACT [Wadden et al., 2020]. We prioritize hardware-constrained example generation, recognizing its significance in fostering accessibility, particularly for companies facing resource constraints. Lastly, we experiment with large language models (LLMs) to explore potential benefits. Mechanically, we measure the accuracy gap reached by an encoder-based fact-checking model when trained on UNOWN examples rather than gold ones.[20] We reveal that the synthetic data produced by million-scale PLMs lead to a gap of only about 2 accuracy points (92.3% ours vs. 94.5% human). When transitioning to

[19]Pronounced as the word "Unknown."

[20]We only changed training data. We used the best models in the target datasets at the start of this study, March 2023, retaining the original implementation and hyperparameters.

TABLE 5.27. Summary of works on the automatic generation of training samples for fact-checking.

Name	U+S*	Domain Agnostic	Tested Datasets†	Human Eval
Pan et al. [2021]	✗	✓	FEVER	✗
Wright et al. [2022]	✗	✗ (biomedical)	SCIFACT	✓
Ours	✓	✓	FEVEROUS SCIFACT	✓

* The study combines unstructured and structured data as evidence.
† FEVER Thorne et al. [2018], SCIFACT Wadden et al. [2020], FEVEROUS Aly et al. [2021].

LLMs, the disparity decreases by one extra point (93.3%). The results, supported by rigorous quantitative and qualitative analysis, attest to the comparable quality between training samples produced by UNOWN and humans, without the need for expertise and labor. In-depth ablation studies provide valuable insights into optimal strategies for the construction of artificial samples, breaking down the impact of each module.

5.4.2 Problem Formulation

5.4.2.1 Fact-Checking Task

Unlike recent studies with a scope confined to tabular evidence [Bussotti et al., 2023], we examine evidence derived from sentences or their combination with tables (i.e., a heterogeneous format). Let $\mathbf{d}$ be a semi-structured document composed of n sentences and m tables, e.g., a Wikipedia page. We define evidence $\mathbf{e} = \{\mathbf{e}_s, \mathbf{e}_t\}$ as a non-empty subset of sentences $\mathbf{e}_s = \{s_1, \ldots, s_{|\mathbf{e}_s|<n}\}$ and, optionally, cell values $\mathbf{e}_t = \{t_i, \ldots, t_{|\mathbf{e}_t|<p}\}$ arising from one table within $\mathbf{d}$, where p is the total cell count. A supervised fact-checker model $\mathcal{F}$ assesses whether a textual claim c finds support or contradiction w.r.t. a given evidence $\mathbf{e}$. Formally, $\mathcal{F}$ takes as input a data pair $<\mathbf{e}, c>$ and outputs a verdict in the space $\mathcal{L} = \{\text{Supports}, \text{Refutes}\}$.[21] Consequently, this work focuses on generating examples $\mathcal{E} = <\mathbf{e}, c, l \in \mathcal{L}>$ for training $\mathcal{F}$.

5.4.2.2 Challenges

UNOWN comprises two steps, as depicted in Figure 5.13. First, $\mathbf{e}$ is created from the input $\mathbf{d}$. Then, $\mathbf{e}$ is used to generate supporting or refuting claims.

Textual Claims There have been research proposals to generate artificial claims by synthesizing $\mathbf{e}$ in a sentence. Abstractive summarization has been explored with text-only evidence [Tonguz et al., 2021, Wright et al., 2022] and scenarios centered on cell values using data-to-text methodologies [Parikh et al., 2020]. In contrast, our goal is to go beyond existing solutions and create claims that can incorporate evidence from

[21] As done in Bussotti et al. [2023], the *Not Enough Information* label is omitted in our analysis due to its infrequency, accounting for only 3% of the instances in FEVEROUS.

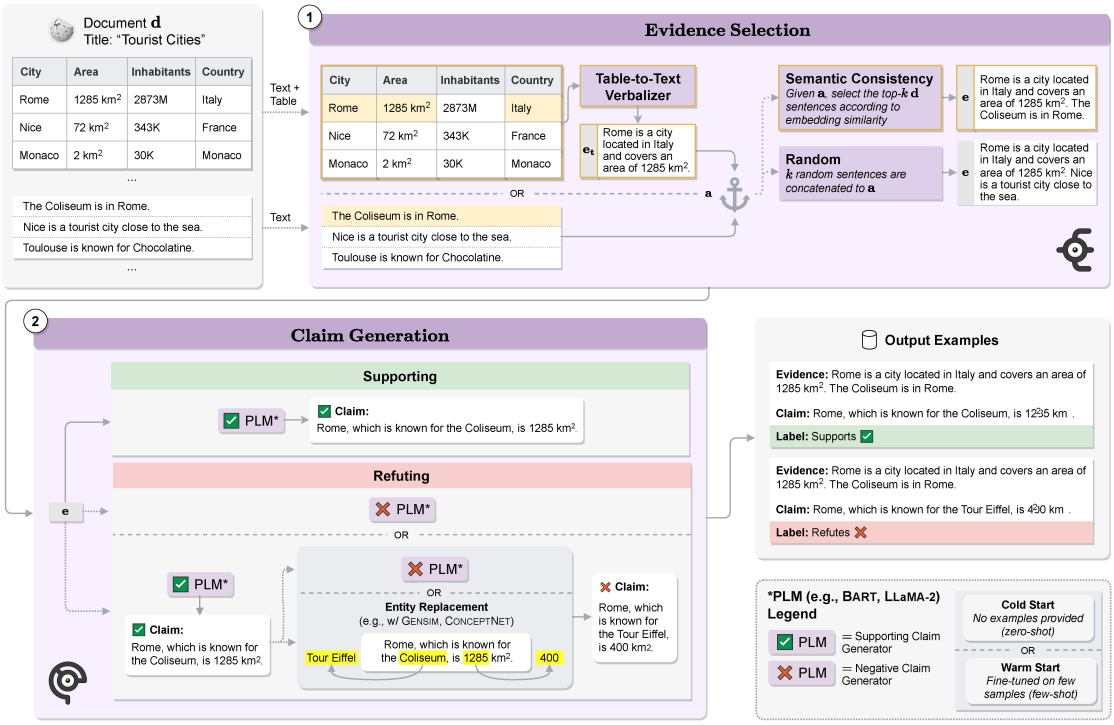


FIGURE 5.14. UNOWN pipeline. The input is a document $\mathbf{d}$ consisting of sentences and optional tables. (1) When both modalities are used, we first define $\mathbf{e_t}$ with a cell sampling and verbalization process. From $\mathbf{e_t}$, different strategies can be used to determine $\mathbf{e_s}$ and complete $\mathbf{e}$. When assuming a text-only approach ($\mathbf{e_t} = \emptyset$), $\mathbf{e}$ is established after sentence sampling. (2) We generate supporting and refuting claims using pretrained language models. We consider text+table evidence with a semantic consistency strategy as our leading example (gold border). Non-continuous lines and arrows delineate alternatives.

structured and unstructured modalities. We emphasize that a *Supports* claim is inherently aligned with the evidence itself. However, to train fact-checking models, we also require examples with a *Refutes* label, which entails claims that are not supported by $\mathbf{e}$. Refuting samples should not be limited to basic negations such as "Rome is not in Italy." They should be adept at capturing nuanced factual contradictions, e.g., "Rome is in France", "There are two Coliseums in Rome." How to obtain such claims is still an open research question.

Flexibility & Scalability Real-world organizations can face obstacles related to low-resource settings, namely commodity hardware infrastructure and poor data availability for training models. These problems, explored with great attention in automatic text summarization [Parida and Motlícek, 2019, Moro and Ragazzi, 2022, 2023, Huh and Ko, 2023, Moro et al., 2023d], affect model supervision and performance. For this reason, fact-checking systems should scale according to user needs to accommodate multiple scenarios. However, in the era of LLMs, the investigation of flexible and scalable solutions is understudied despite their high social impact [Tamkin et al., 2021].

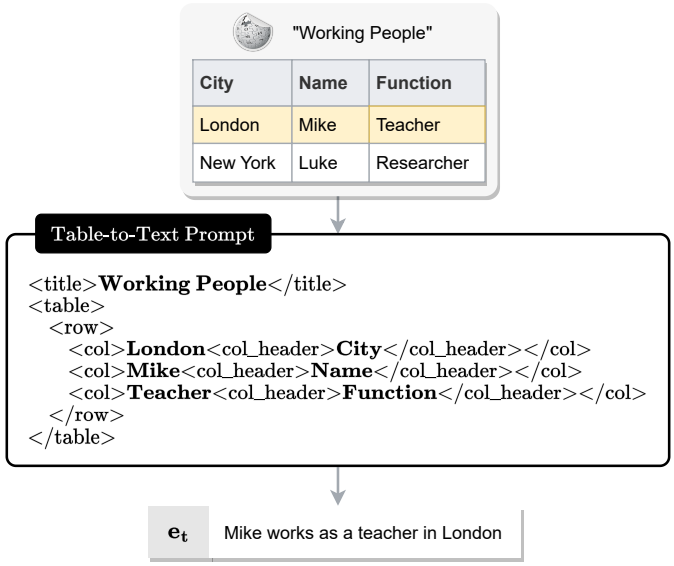


FIGURE 5.15. Natural language verbalization of a subset of tabular cells (yellow background).

5.4.3 Method

We introduce UNOWN (Figure 5.14), a novel framework to automate the production of fact-checking training data. UNOWN consists of two main tiers: evidence selection (Section 5.4.3.1) and claim generation (Section 5.4.3.2) stages.

5.4.3.1 Evidence Selection

The evidence construction process unfolds iteratively, starting with a textual anchor **a**.

- *Text-only.* We initialize **a** by randomly selecting a sentence from **d**.
- *Text+table* (i.e., tabular data are accessible). We determine **a** by collapsing the two modalities. A recent trend seeks to unify various sources of information using text as a standardized representation Berrios et al. [2023], Tan et al. [2023], Zeng et al. [2023]. In alignment with this text-centric vision, we replicate the work of Veltri et al. [2023] and fine-tune a T5 model [Raffel et al., 2020] on TOTTO [Parikh et al., 2020], a table-to-text dataset. Specifically, we sample cells[22] from a table and generate **a** by inference. The prompt is formulated using cell values while maintaining their contextual details, including the table headers and **d** title (Figure 5.15). A consistent text mode streamlines the claim generation stage, but still leaves the question of how to select evidence from **d**.

Once **a** is designated, we propose two alternative strategies to complete the evidence.

- *Semantic consistency.* $\mathbf{e_s}$ is crafted by concatenating the k sentences from **d** that semantically align the most with **a**, thus preserving the topic coherence. Drawing

[22] Cell extraction is a consolidated practice for evidence retrieval in table-based factuality predictors Aly et al. [2021].

inspiration from Liu et al. [2023], we consider cosine similarity after T5 encoding.
- *Random.* We pick k random sentences from $\mathbf{d}$. Various topics may exist within $\mathbf{e}$, as the information chosen may not be aligned.

We conclude the discussion on evidence selection with some clarifications. **(1)** In text-only scenarios, $\mathbf{e_t}$=$\emptyset$ and $\mathbf{e}$ consists solely of a set of $\mathbf{d}$ sentences. In text+table scenarios, $\mathbf{e}$ comprises $\mathbf{d}$ sentences and a verbalized representation of tabular cells. We overwrite $\mathbf{e}$ by prefixing the $\mathbf{d}$ title for context with special `<title>` and `<evidence>` token delimiters. Basic concatenation enables cross-attention among the page title, cells, and sentences. **(2)** In both settings, k is drawn randomly from $[1, 1, 2, 2, 2, 3, 3, 4, 5]$, a distribution derived from the analysis of the FEVEROUS train set. **(3)** We stress that $\mathbf{e_t} \rightarrow \mathbf{e_s}$ is the most natural sense of evidence construction that involves a single verbalization operation. The reverse approach would be impractical, mainly considering the initial $\mathbf{e_t}$ cell-level granularity.

In the remainder of this section, we deal with text-only data, as all evidence $\mathbf{e}$ is in such a form.

5.4.3.2 Claim Generation

First, we delve into the role of existing labeled examples, when available. We then present our synthesis procedure, using small models that can run on laptop-grade GPUs or Google Colab. The discussion ends with an examination of LLMs.

Warm & Cold Start In a practical setting, users can reasonably expect access to supervised data from related fact-checking applications. Hence, we use the term *warm start* to describe a preliminary fine-tuning of PLMs on existing labeled examples ($\mathbf{e} \rightarrow c$). Aligning with real-world conditions, we posit that the number of samples should not exceed 100. Differently, we use the term *cold start* to denote the absence of related data.

Small Models State-of-the-art million-scale PLMs, here referred to as small models, have the potential to provide an accessible and efficient solution for claim generation. In the warm-start phase, we exploit pre-existing examples to get the most out of the selected model. Fine-tuning is performed on a few instances of external datasets that exhibit the same $<\mathbf{e}, c, l>$ structure as our generation goal. We also explore training with a limited portion of fact-checking benchmarks (train set). In both cases, human-created evidence and claims act as input and gold output.

Refuting Claims Generating refuting claims comes with additional intricacies. We recognize two main paths to avoid introducing a strong lexical bias in artificial training samples, such as basic and homogeneous negation types.
- *Direct refusal.* In a cold-start scenario, we employ a PLM already capable of translating an input evidence text $\mathbf{e}$ into a refuting claim. We extend the PLM training to extra refuting examples for a warm start.

Claim Generation Prompt

Write a claim that uses the following evidence.
Write a negative claim, i.e., false with regard to the following evidence.

Evidence:
<title> {{**d** title}} <evidence> {{**e**}}

Claim:
y

FIGURE 5.16. Instruction tuning prompt template for claim generation. Only the highlighted part is used for loss computation. Green and red colors denote alternative instructions for supporting and refuting targets, respectively.

- *Two-step approach.* We summarize **e** into a supporting claim, then undergo a methodological ramification to alter its meaning.
  - *Direct refusal.* The same as before, but with the supporting claim as input.
  - *Entity replacement (ER).* We randomly substitute words with antonyms or related terms from a knowledge base.

LLMs LLMs offer an alternative to small models, either with zero-shot prompting (cold start) or instruction tuning (warm start). Our template is shown in Figure 5.16.[23] During fine-tuning, the LLM is presented with **e** and trained to produce an artificial claim according to an instruction specifically geared toward supporting or refuting.

5.4.4 Experimental Setup

Research Questions We conduct thorough experiments to answer the following queries.

- Q1 Are the generated examples good enough for training the target model?
- Q2 What is the best strategy to select evidence?
- Q3 What is the recommended method to generate refuting claims?
- Q4 Does the efficacy of synthetic examples hold in various fact-checking domains?
- Q5 How many human-annotated samples are required to bootstrap the downstream fact-checking model effectively? Can a fully unsupervised solution rival in competitiveness?

Datasets In warm-start regimes, we take human examples from FEVER [Thorne et al., 2018], a collection of claim–evidence pairs grounded on Wikipedia. As the leading fact-checking benchmark, we take FEVEROUS [Aly et al., 2021], an extension of FEVER with

[23] We undertook exhaustive prompt tuning. Preliminary experiments proved the non-indispensable role of few-shot in-context learning. We then opted for a simpler and reproducible zero-shot approach, also more fair to small models.

TABLE 5.28. Dataset statistics.

Dataset	Use Case	Veracity Labels	Claim Length	Evidence Sent./Cells*
FEVER	Train	10K ✅ / 10K ❌	8.1	2.4/0
FEVEROUS 🗎	Test	1.5K ✅ / 1.7K ❌	27.1	2.1/0
FEVEROUS 🗎 + ⊞	Test	1.5K ✅ / 0.5K ❌	25.2	1.6/4.4
SCIFACT	Test	0.2K ✅ / 0.1K ❌	12.3	1.8/0

* Average.

more complex claims enriched with structured evidence. To gauge the generality of our framework, we regard SCIFACT [Wadden et al., 2020], a dataset of expert-written claims paired with evidence selected from abstracts of scientific papers. Distributional statistics are given in Table 5.28.

Metrics We assess fact-checking predictions using Accuracy and F1 scores ($[0, 1]$; the higher, the better), distinguishing between *Supports* and *Refutes* labels. Precisely, we validate models on the test set of downstream datasets after a training phase on varying-size collections of artificial examples. To incorporate an intermediate signal, we assess the logical relationship between each evidence–claim pair. To this end, we apply a NLI-pretrained DEBERTA model to infer *Entailment*, *Contradiction*, or *Neutral* labels.[24]

Claim Generation Models We use small models built on BART [Lewis et al., 2020a], an encoder-decoder PLM featuring denoising pretraining. For supporting claims, we scrutinize the large version (400M parameters). For refusing claims, we contemplate BART-large and a specialized BART-base model (140M parameters), here termed BARTNEG [Lee et al., 2021b], trained on parallel and opposing claims from the English dataset WIKIFACTCHECK [Sathe et al., 2020].[25] We stress that refusal can be obtained by: (i) querying these models directly on $\mathbf{e}$; (ii) applying these models or ER methods on the claim returned by a supporting model. As LLM, we operate with LLAMA-2 7B [Touvron et al., 2023b] (decoder-only), opting for QLORA adapter fine-tuning. Fine-tuning is done independently for the two claim types. To train BART, we set learning rate=$1e^{-4}$, batch size=16, and number epochs=20; for LLAMA-2, we use 4-bit nested quantization, r=8, α=32, batch size=1, and number epochs=3. At inference time, we adopt beam search (num_beams=5) and nucleus sampling (top_p=0.01, top_k=40, temp=0.15) for BART and LLAMA-2, respectively.

Entity Replacement Methods We adapt the pipeline proposed by Wright et al. [2022] to domain-agnostic resources, studying three alternative refusal procedures.

[24] The model is available in HuggingFace: `cross-encoder/nli-deberta-v3-base`

[25] Despite the BARTNEG first-stage fine-tuning, we point out that WIKIFACTCHECK's claims are substantially more inclined toward real-world scenarios, i.e., longer statements characterized by a sophisticated language, crawled from the wild rather than being curated by annotators. Moreover, BARTNEG is trained on transitioning between single-sentence claims, justifying the need for a warm start.

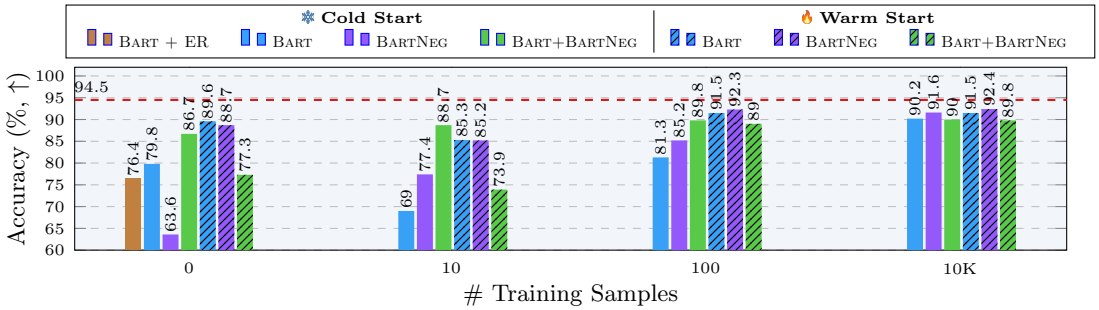


FIGURE 5.17. The overall accuracy scores on FEVEROUS by varying the number of training samples. The dashed bars indicate leveraging a fine-tuning on FEVER. The horizontal red dashed line represents the accuracy obtained by human data.

- FLAN-T5. `flan-t5-large` [Wei et al., 2022a] is prompted by a conversation style: *"Answer the following question. Can you give me an antonym of* {{**w**}}*?"*, where **w** is a word of length ≥ 4 randomly chosen for replacement.
- GENSIM. The GENSIM library [Rehurek and Sojka, 2011] (`glove-twitter-25` model) is used to calculate a similarity matrix between the words in the supporting claim. The matrix is subsequently used to build a frequency ranking, aiding in deciding which word to replace (least common, most common, random). Denoting the chosen item as **w**, words having similarity > 0.7 to **w** are substituted with a similar but distinct word as per WORDNET [Miller, 1995].
- CONCEPTNET. `conceptnet.db` [Speer et al., 2017] is used to identify a set of closely related concepts for each word in the claim. We assess the frequency ranking by examining the intersection of concept lists. Similarly to GENSIM, we set a concept and replace similar words based on `antonym` relationships or, in their absence, `is-a`.

Fact-Checking Models We fairly assess the impact of our synthetic training examples on accurately predicting the verdict label of an input claim given a set of evidentiary sentences. To achieve this, we choose optimal classification models for the two benchmarks at hand, keeping their weights and hyperparameters unchanged. For FEVEROUS, we utilize the ROBERTA encoder [Liu et al., 2019] with a linear layer on top. For SCIFACT, we prefer MULTIVERS [Wadden et al., 2022], a LONGFORMER encoder [Beltagy et al., 2020] with a shared representation of the claim and the evidence-containing scientific abstract.

Environment We run each experiment on a cluster of OS Linux workstations, using a single Nvidia GeForce RTX3090 Turbo GPU of 24 GB VRAM. UNOWN is developed using PyTorch [Paszke et al., 2019] and the HuggingFace library [Wolf et al., 2019], setting the seed to 42 for the sake of reproducibility.

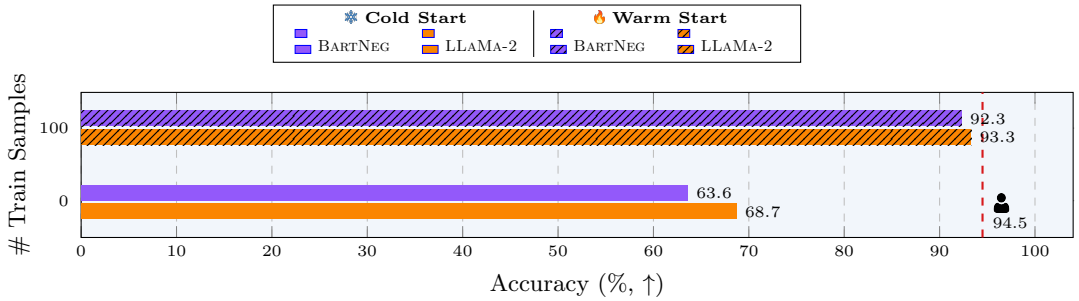


FIGURE 5.18. Accuracy scores on FEVEROUS with training examples generated by LLAMA-2.

5.4.5 Results and Discussion

5.4.5.1 Quality of Generated Claims

Concentrating on Q1, we measure how well the UNOWN examples generated with small models contribute to training a downstream fact-checking system on the FEVEROUS test set (Figure 5.17). In the worst-case scenario (cold-start, zero-shot), the highest accuracy achievable by UNOWN is 86.7 with BART-large for both supporting and refuting claims. The results portray a consistent boost in performance with an expansion of target training instances. In fact, the accuracy climbs to 92.4 with warm start and 10K target examples, using BART-large for supporting claims and BARTNEG for direct refusal—very close to the accuracy achievable with human-authored data (94.5). Although having 10K human examples in real-world fact-checking scenarios may be a stretch, training the same model configuration with 100× fewer examples causes a marginal 0.1 drop in accuracy (92.3).

Figure 5.18 looks at how the claims generated by LLAMA-2 stack up against those inferred by the best small model setup. The accuracy propelled by LLAMA-2 claims, following training on 100 examples, stands at 93.3, outperforming BART-large and BARTNEG by a single point. Hence, incorporating LLMs appears non-essential in the UNOWN pipeline, favoring BART-based models for their superior effectiveness-efficiency trade-off (see Table 5.29 for additional efficiency statistics).

TABLE 5.29. Time consumption for different tasks.

Model	Task	sec/Claim
Claim Generation		
BART	`Train/Infer.`	1.92 / 0.12
BARTNEG	`Train/Infer.`	1.01 / 0.08
LLAMA-2	`Train/Infer.`	1.98 / 2.10
Table-to-Text		
T5-TOTTO	`Infer.`	0.75
Evidence Selection (Semantic Consistency)		
T5	`Tokeniz.` + `Distance`	5.43

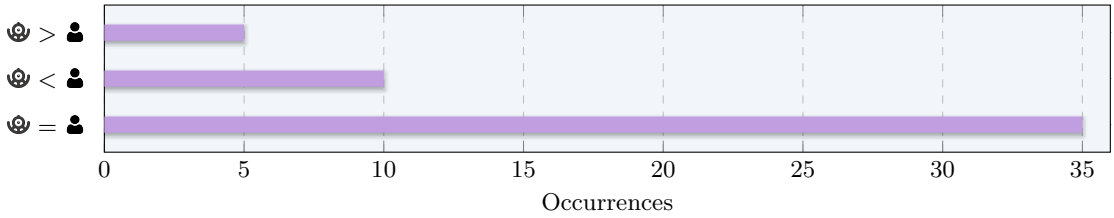


FIGURE 5.19. Human evaluation results on 50 claims.

We undertake a qualitative analysis to delve deeper into the quality of generated claims. We sample 50 instances from FEVEROUS training data (25 supporting, 25 refuting). Considering the expense associated with careful human evaluation and the central role of text as our unified modality, we accorded priority to text-only evidence. Each instance comes with human-selected evidence and the corresponding claim. We condition on the original evidence to ensure fairness and use UNOWN to produce synthetic claims with our best models: warm-started BART-large and BARTNEG. Afterward, we provide three human annotators – chosen among the authors, with evidence and two claims in randomized order, avoiding categorizing the latter as "original" or "generated" (i.e., blind review). Motivated by Narayan et al. [2018], Fabbri et al. [2019], and Moro et al. [2023e], we use a direct comparison assessment that has been shown to be more reliable, sensitive, and less labor intensive than rating scales. Indeed, we ask the annotators to state which claim is the best w.r.t. two dimensions: *clarity* (effective communication of the intended meaning with a good sentence structure, fluency, and English precision) and *coherence* (semantic connection to the evidence). A tie is declared if a judge perceives the two claims to be of comparable quality. We gather the replies and aggregate them using a majority voting approach. The Cohen's κ coefficient, indicating the agreement between each annotator and the label derived from majority voting, is 0.613, signifying a commendable level of agreement. Figure 6.29 illustrates that, of the 50 paired claims, humans find 35 to be of comparable quality. In 10 cases, the original FEVEROUS claims are deemed superior, whereas in 5 cases, the examples generated by UNOWN are evaluated as better than those crafted by humans.

Figure 5.20 reports input–output claim generation examples on top of FEVEROUS, accompanied by human evidence and claims. Even if the artificial claims vary, they all present a correct syntax and proper use of the given evidence.

Overall, the generated examples prove sufficiently effective for training fact-checking models, yielding quantitative results within a 2-point margin in absolute accuracy compared to those achieved by a crowd of human annotators.

5.4.5.2 Evidence Selection

We now discuss Q2 by studying the impact of alternative evidence selection methods. We design two experiments using FEVEROUS training examples: one with text-only evidence and another with text+table evidence; test datasets are filtered according to the scenario. For every human example, referred to as "gold", we execute our best BART

Evidence (d title = Windecker Eagle)	
The Eagle AC-7 Eagle 1 (USAF designation YE-5) is an aircraft that was manufactured by Windecker Industries Data from Jane's All the World's Aircraft 1971.	
☑ Supporting Claims	
HUMAN	The Eagle AC-7 Eagle 1 is a military aircraft that was manufactured by Windecker Industries.
LLAMA-2	Windecker Eagle is listed in Jane's All the World's Aircraft 1971-72.
BART	In 1896, Friedrich Baume established a legal practice with A. E. Whitaker, a son of Frederick Whitaker, and won the election.

Evidence (d title = Stefan Lessard)	
Lessard was born in Anaheim, California, to musician Ron and Jacqueline Lessard. During the 2007 European Tour, Lessard *also began* a video blog on YouTube. In October 2005 he married Jaclyn Burton, and together they have *three daughters*, Hazel Bailey (born March 6, 2004), Flora Jude (born July 13, 2006), and India (born November 2015).	
✖ Refuting Claims	
HUMAN	Stefan Kahil Lessard (born on June 4, 1974 outside Anaheim, California) completed his junior year of high school, passed the GED examinations in the summer of 1991, and was accepted into the Jazz Studies program at Virginia Commonwealth University.
LLAMA-2	Stefan Lessard has been married to Jaclyn Burton since October 2005. They have **three sons** named Hazel Bailey (born March 6, 2004), Flora Jude (born July 13, 2006) and India (born November 2015).
BART	Stefan Lessard **never began** a video blog on YouTube.

FIGURE 5.20. Qualitative examples on claim generation, highlighting negation artifacts (bold) w.r.t. evidence excerpts (italic underline).

model with four alternative evidence selection strategies. Human evidence, where we use the original evidence handpicked by the FEVEROUS annotators. Random with gold, where the number of selected sentences matches the human example, but the actual cells and sentences are chosen randomly from $\mathbf{d}$. Random without gold, where the number of retrieved sentences k, after anchor definition, is drawn from the distribution presented in Section 5.4.3.1. Semantic consistency.

Table 5.30 documents Accuracy and F1 results. The influence of evidence is evident. The use of human evidence allows UNOWN to produce examples that nearly matches

TABLE 5.30. Evidence selection comparison in FEVEROUS. All models use BART and BARTNEG to generate supporting and refuting claims, respectively. Both models are first fine-tuned on FEVER and then on 100 FEVEROUS instances.

	Feverous			
	Method	**Acc.**	**F_1 ☑**	**F_1 ✖**
	`Text`			
👤	HUMAN EVIDENCE + CLAIM	94.50	94.92	95.30
👤 + ⚙	HUMAN EVIDENCE	92.40	92.10	92.70
	RANDOM W/ GOLD $\lvert\mathbf{a}\rvert + k$	92.30	91.70	92.70
⚙	RANDOM W/O GOLD $\lvert\mathbf{a}\rvert + k$	85.11	86.07	84.01
	SEMANTIC CONSISTENCY	88.33	88.21	88.44
	`Text + Table`			
👤	HUMAN EVIDENCE + CLAIM	82.09	87.83	66.12
👤 + ⚙	HUMAN EVIDENCE	84.59	90.53	58.69
	RANDOM W/ GOLD $\lvert\mathbf{a}\rvert + k$	81.81	87.96	62.89
⚙	RANDOM W/O GOLD $\lvert\mathbf{a}\rvert + k$	84.43	89.90	66.03
	SEMANTIC CONSISTENCY	86.83	91.73	67.65

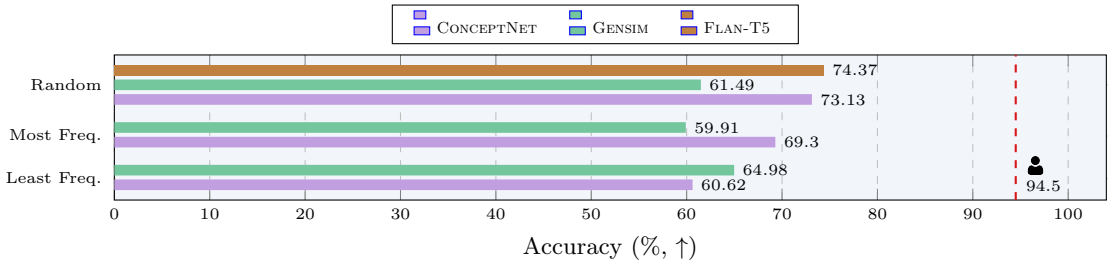


FIGURE 5.21. Comparison of different entity replacement methods in FEVEROUS.

the human upper bound. Notably, in the text+table setting, we reach even higher scores for supporting claims, ascertaining the quality of our claim generator. In the text-only scenario, performance is optimal when guided by the cardinality of human gold evidence, with random selection surpassing semantic consistency. In text+table, semantic consistency outperforms both random selection and original human examples in all metrics. We observe that human annotators struggle to annotate tabular data accurately, thus making mistakes that mislead the classifier. This is also reflected in the generally lower results for text+table compared to the text-only scenario.

5.4.5.3 Refuting Claims

We debate Q3 by showing how the fact-checking performance varies with different types of *Refutes* generated claims. We report on a quantitative analysis and then a qualitative one from a user study.

Quantitative FLANT5 stands out as the best ER method, as depicted in Figure 5.21; unless otherwise specified, we use ER to denote this approach. Figure 5.17 includes the impact of various negation strategies on the accuracy of the target task. In cold start, the combination of BART and BARTNEG using the two-step approach is effective, while the results are subpar with ER. ER falls short in claim refusal, possibly due to limitations in content replacement without appropriate rewording. As anticipated, initiating with a warm start is advantageous, resulting in the highest accuracy with 0 and 100 training samples.

To gain additional insight into the generated claims, we compute the NLI prediction score between the claim and the evidence. Table 5.31 shows that, for supporting claims, UNOWN's examples closely resemble the score distribution in their human-written counterparts. Yet, in the refuting examples generated by UNOWN, the percentage of entailed claims surpasses that of human-generated ones. This confirms the difficulty in creating refuting examples compared to supporting ones. Once again, ER appears to be poor in effectiveness.

Qualitative We perform a human analysis to evaluate the negation techniques employed in refuting claims. We adhere to the negation taxonomy outlined in previous

TABLE 5.31. The quality of the generated claims based on NLI scores (text-only).

	Feverous			
	Method	**Entail.**	**Contrad.**	**Neutral**
	`Supporting`			
	HUMAN	75.00	4.00	21.00
	BART	74.57	4.90	20.53
	LLAMA-2	71.92	3.95	24.12
	`Refuting`			
	HUMAN	3.00	77.00	30.00
	BART	10.43	56.00	33.57
	LLAMA-2	11.23	37.82	50.95
	ENTITY REPLACEMENT	36.05	40.70	23.26

studies [Zafra et al., 2020, Dobreva and Keller, 2021]. Rigorously, we use two main negation types (*Verbal Negation* (V) and *Noun Phrase Negation* (NP)), each classifiable in three sub-classes. The subclasses include *Lexical* (L), where the negation is expressed with new words or phrases that alter the sentence meaning (e.g., 10 papers→**more than** 10 papers), *Morphological* (M), where the form of the word is modified through morphemes (e.g., legal→**il**legal), and *Replacement* (R), where a phrase is swapped for another with a different meaning (e.g., 1995→1997). Given these classes, three annotators (selected among the authors) evaluated 30 claims from the FEVEROUS human-defined train dataset and 30 claims generated by UNOWN. The final category is identified by majority voting over the three suggested labels; the Cohen's κ coefficient is 0.91, which shows very high agreement among annotators. The results of the study are illustrated in Figure 5.22, enabling a comparison of annotation distributions between the two sets of examples (UNOWN vs human). UNOWN produces an even distribution of refuting claims, encompassing both noun phrases and verbal structures, whereas humans tend to prefer noun phrases. Both UNOWN and humans evidently favor the substitution strategy for noun phrases and the lexical strategy for verbs. In both scenarios, the ranking of classes and subclasses remains consistent, indicating that UNOWN can produce a range of negation types comparable to those observed in a corpus crafted by humans.

5.4.5.4 Checking Scientific Claims

We discuss Q4 by measuring the quality of the fact-checking system trained with our generated examples in a different domain. To this end, we use the scientific dataset SCIFACT. Table 5.32 confirms the main results from the analysis done over FEVEROUS data. The human training data achieves the best results, followed by UNOWN with the (BART) warm start. Again, BARTNEG leads to better results w.r.t. ER. We explain the bigger gap in the results between human and UNOWN because the warm start includes only examples from FEVER and no ones from SCIFACT.

	SciFact			
	Method	**Accuracy**	**F_1 ✅**	**F_1 ❌**
	Human Training Data	84.62	81.05	85.71
	Entity Replacement	55.33	57.27	16.51
	BartNeg	65.08	53.79	65.96
	Entity Replacement	54.00	57.63	1.98
	BartNeg	74.50	73.53	73.45

Table 5.32. Fact-checking results on SciFact with different strategies for refuting claim generation. All models use Bart to create supporting claims. In warm scenarios, models are trained on Fever.

5.4.5.5 Cold vs. Warm Start

We discuss Q5 by measuring the impact of the examples used to fine-tune the models. As shown in Figure 5.17, Figure 5.18, and Table 5.32, a warm-start approach is key to improving the quality of the generated data. More precisely, Figure 5.23 shows the average Δ accuracy improvement when shifting from cold to warm in Feverous. We observe a decrease in performance as the number of samples from the target dataset increases, highlighting the ameliorative contribution of using external related data as a guiding source of knowledge. Also SciFact exhibits increased accuracy for BartNeg in the warm approach.

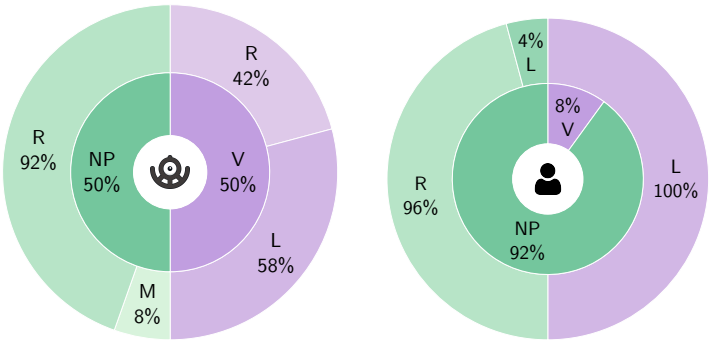


Figure 5.22. Human annotation on negation artifacts in refuting claims.

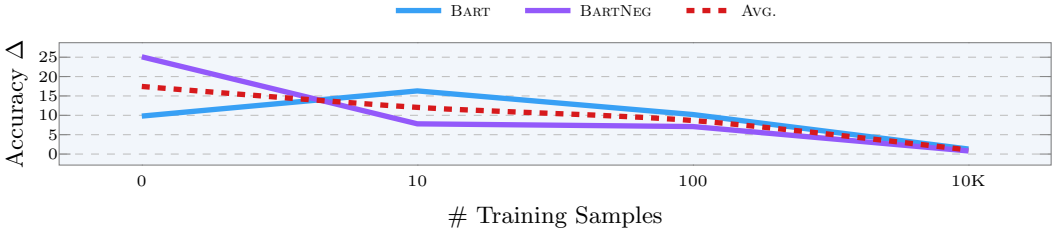


Figure 5.23. The Feverous's average Δ accuracy improvement when shifting from cold to warm.

5.4.6 Related Work

5.4.6.1 Computational Fact-Checking

Fact-checking has been a focal point in NLI for decades [Dagan et al., 2005, Guo et al., 2022b]. The emergence of LLMs and services such as ChatGPT [Schulman et al., 2022] has facilitated the establishment of fact-checking pipelines. However, initial attempts to use such models have revealed their inefficacy compared to human experts [Saeed et al., 2022, Caramancion, 2023]. Currently, adopting specialized alternatives is the most viable strategy, despite the demand for large labeled datasets for training models.

5.4.6.2 Example Generation

The creation of fact-checking training examples has been approached through both unsupervised and supervised methodologies. *Unsupervised* solutions, typically employed when labeled samples are unavailable, utilize PLMs to generate textual claims, e.g., using template prompts to derive claims from a given text [Meng et al., 2022a]. *Supervised* methods rely on specific resources, e.g., an annotated taxonomy to train a Lstm model for sentence generation [Meng et al., 2019].

Textual Evidence Several works focus on generating claims from unstructured data. Pan et al. [2021] produce question–answer pairs from textual evidence, using answer replacement to assemble the refuting claim. Wright et al. [2022] create supporting claims using a seq2seq PLM and entity replacement over a domain-specific KB for evidence refusal in the biomedicine field.

Tabular Evidence Few papers focus on obtaining claims from tabular data. One approach reports on a template-based solution [Veltri et al., 2023], while another shows that a query generation solution from human provided examples leads to better results [Bussotti et al., 2023]. In the latter, the idea is to infer queries from the examples to fetch new evidence from other tables. The evidence is then fed to an LLM in a few-shot prompt to generate textual claims. Similarly to Feverous, a SemEval-2021 Task [Wang et al., 2021] consists of 2K tables in which claims are built for fact verification and cell evidence selection. From tables, claims are generated using IBM Watson Discovery based on templates. None of the existing methods consumes unstructured and structured data as input for generating training examples.

5.4.7 Conclusion

We introduced Unown, a novel domain-agnostic framework to automatically generate training examples for fact-checking systems, bypassing the costly task of manually annotating large volumes of data. Unown fits both structured and unstructured data to compile textual claims that support or refute the evidence provided. It also accommodates several solutions for evidence selection and claim generation to adapt to different requirements, such as the lack of high-memory GPUs. We evaluated our framework

using two publicly accessible datasets that deal with general-domain and scientific contexts. The results suggest that our synthetic examples exhibit a quality comparable to that of expert-labeled data, indicating the practicality and efficacy of our framework.

Although UNOWN is a promising step forward, some research directions remain untapped. Primarily, our generation process lacks coverage of certain examples within the long tail, e.g., mathematical operations, such as the premise "Paul is 2 years younger than Mary." We consider using a solution in which more intricate patterns are generated as queries over relational tables [Veltri et al., 2023]. Secondly, once models have been trained with instances from UNOWN, we could set up active learning algorithms to guide our methods to generate examples that effectively enhance performance on the test set [Zhang et al., 2022e].

Chapter 6

Datasets and Benchmarks

6.1
Carburacy: Summarization Models Tuning and Comparison in Eco-Sustainable Regimes with a Novel Carbon-Aware Accuracy

Originally published as: *Gianluca Moro, Luca Ragazzi, and Lorenzo Valgimigli. Carburacy: Summarization models tuning and comparison in eco-sustainable regimes with a novel carbon-aware accuracy. In Brian Williams, Yiling Chen, and Jennifer Neville, editors, Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023, pages 14417–14425. AAAI Press, 2023b. doi: 10.1609/AAAI.V37I12.26686. URL https://doi.org/10.1609/aaai.v37i12.26686*

Abstract

Generative transformer-based models have reached cutting-edge performance in long document summarization. Nevertheless, this task is witnessing a paradigm shift in developing ever-increasingly computationally-hungry solutions, focusing on effectiveness while ignoring the economic, environmental, and social costs of yielding such results. Accordingly, such extensive resources impact climate change and raise barriers to small and medium organizations distinguished by low-resource regimes of hardware and data. As a result, this unsustainable trend has lifted many concerns in the community, which directs the primary efforts on the proposal of tools to monitor models' energy costs. Despite their importance, no evaluation measure considering models' eco-sustainability exists yet. In this work, we propose CARBURACY, the first carbon-aware accuracy measure that captures both model effectiveness and eco-sustainability. We perform a comprehensive benchmark for long document summarization, comparing multiple state-of-the-art

quadratic and linear transformers on several datasets under eco-sustainable regimes. Finally, thanks to CARBURACY, we found optimal combinations of hyperparameters that let models be competitive in effectiveness with significantly lower costs.

6.1.1 Introduction

In the last few years, we have witnessed remarkable progress on a broad range of natural language processing (NLP) tasks [Borgeaud et al., 2022, Moro et al., 2022a, Frisoni et al., 2023], accomplished by increasingly large and computationally-hungry transformer-based models [Floridi and Chiriatti, 2020, Smith et al., 2022] backed by high-performance dedicated hardware such as GPUs and TPUs [Wang et al., 2019c]. Such solutions are proposed to obtain new state-of-the-art results, namely enhancing the score of automatic evaluation measures that focus on effectiveness (e.g., `ROUGE` [Lin, 2004] for text summarization and `F1` for question answering), completely overlooking the economic, environmental, and social costs of attaining those outcomes [Strubell et al., 2020]. Thus, the AI community started raising concerns for this unsustainable trend [Bender et al., 2021], proposing tools to monitor energy costs without concrete solutions [Henderson et al., 2020].

An example of an NLP task affected by this shift is long document summarization, which aims to compress a lengthy input text into a shorter version while preserving the salient details in terms of informativeness and factual consistency [Koh et al., 2023a]. This task is dominated by large solutions [Wu et al., 2021a, Mao et al., 2022] requiring massive labeled training data and high-memory GPUs (up to 48 GB) to obtain state-of-the-art results at the expense of a high carbon footprint [Dhar, 2020, Patterson et al., 2022]. Consequently, this untenable tendency narrows the research to large companies, raising barriers to small and medium organizations distinguished by low-resource regimes (e.g., GPUs of 12 GB memory and 100 training samples). Besides, in the face of the climate change problem, it is compulsory to minimize carbon sources to avoid rising global temperatures [Pörtner et al., 2022]. Hence, we argue that the research must move toward environmental-friendly cutting-edge approaches [Tambe et al., 2021, Du et al., 2022, Moro and Ragazzi, 2022, Frisoni et al., 2022c, Moro et al., 2023d]. Nevertheless, there is no automatic evaluation measure to capture the eco-sustainability of models, restricting the rise of new research directions toward greener solutions.

This paper presents CARBURACY,[1] the first carbon-aware accuracy measure to evaluate models by considering the performance and the CO_2 emissions demanded to fulfill their effectiveness score. The key features of CARBURACY are: (1) Applicability in multiple NLP tasks, namely those assessed by metrics that produce a score $\in [0, 1]$ (e.g., text summarization and classification). (2) Two hyperparameters used to tune what should be prioritized and rewarded (i.e., model effectiveness or eco-sustainability). To test CARBURACY, we benchmark multiple state-of-the-art quadratic and linear transformers on various long document summarization datasets under challenging scenarios of limited hardware and data. Precisely, we study the impact of many hyperparameters

[1] The code is at https://github.com/disi-unibo-nlp/carburacy

on model effectiveness and eco-sustainability, optimizing the pipelines toward greener training. Moreover, we investigate the inference phase, which is paramount for model deployment at the business level.

Contributions Our contributions are the following:
(1) *A novel carbon-aware accuracy measure to automatically consider eco sustainability.* In this work, we propose CARBURACY, which fits many NLP tasks and assesses models based on the cost needed to achieve their effectiveness.
(2) *In-depth comparison of models in long document summarization under low-resource regimes.* We perform a deep benchmark of state-of-the-art models in long document summarization under challenging settings of hardware and data.
(3) *Findings of optimal combinations of hyperparameters.* Thanks to CARBURACY, we uncover the hyperparameters that let models obtain high effectiveness with low costs.

6.1.2 Related Work

Carbon Emissions Measures Several works presented tools to estimate the energy consumption [Yang et al., 2017] and carbon footprint [Anthony et al., 2020, Naidu et al., 2021] of machine learning models and transformers [Cao et al., 2021, Lal et al., 2021], fighting against the problem of climate change by promoting future directions toward energy-efficient solutions. Despite their importance, no single evaluation measure considering the effectiveness and cost has been proposed, confining the development of new environmentally friendly state-of-the-art models.

Energy Costs Comparative Studies Various works proposed studies on the environmental costs of models, ranging from computer vision [Li et al., 2016, Hampau et al., 2022] to NLP [Cao et al., 2020, Strubell et al., 2020, Bannour et al., 2021, Zhou et al., 2021]. Nonetheless, there is no comparative analysis of models in long document summarization and under low-resource scenarios, ignoring the many different infrastructures that research teams and organizations may have.

Our Work Unlike prior contributions, our paper pioneers the exploration of the first carbon-aware accuracy measure (CARBURACY), presenting an in-depth comparative analysis of cutting-edge models in long document summarization under low-resource regimes of hardware and data. Accordingly, CARBURACY is not directly comparable with existing cost estimation metrics because it is the first measure that integrates both costs and effectiveness in a single score.

6.1.3 Methodology

In this section, we define the concept of effectiveness and cost and introduce our novel measure dubbed CARBURACY.

TABLE 6.1. Effectiveness formulas for different NLP tasks. Some tasks miss a custom formula because they are already evaluated with a single metric that produces a score $\in [0, 1]$.

Task	Eval Metric	Effectiveness ($\mathcal{R}$)
Summarization	ROUGE (r_1, r_2, r_L)	$\frac{\mathcal{A}(r_1, r_2, r_L)}{1 + \sigma_r^2}$
Translation	BLEU	BLEU
Classification	Accuracy / F1	Acc / F1 / $\frac{\mathcal{A}(\texttt{Acc}, \texttt{F1})}{1 + \sigma^2_{\texttt{Acc,F1}}}$

6.1.3.1 Effectiveness

The effectiveness ($\mathcal{R}$ henceforth) is the result in terms of the model's accuracy. In long document summarization, $\mathcal{R}$ is evaluated with ROUGE [Lin, 2004], which reckons the lexical overlaps of words and sentences between the inferred and target summary with three scores (i.e., r_1, r_2, r_L). For text summarization, we formally define $\mathcal{R} \in [0, 1]$ as:

$$\mathcal{R} = \frac{\mathcal{A}(r_1, r_2, r_L)}{1 + \sigma_r^2} \tag{6.1}$$

where $\mathcal{A}$ is the average function and σ_r^2 is the variance of the ROUGE scores. By dividing the averaged score by the variance, two models with the same average but different scores are not considered equal. Indeed, high variance means that the generated summary fails on some scores. For instance, our measure considers a model that obtains an $\{r_1, r_2, r_L\}$ score of $\{0.4, 0.2, 0.3\}$ better than a model that achieves $\{0.5, 0.15, 0.25\}$, despite having the same averaged score.

Equation 6.1 explicitly defines the effectiveness of those tasks estimated with ROUGE, but it can be formulated for numerous NLP tasks (Table 6.1), e.g., text classification [Moro et al., 2018] and information retrieval [Moro and Valgimigli, 2021].

6.1.3.2 Cost

The cost ($\mathcal{C}$ henceforth) denotes the resources required for a model to obtain $\mathcal{R}$. A first cost formulation is defined in Schwartz et al. [2020] as $\mathcal{C} \propto E \cdot D \cdot H$, where E is the cost of processing a single example, D is the size of the dataset, and H is the number of experiments for hyperparameter search.

Since we aim to investigate the correlation between the model effectiveness and cost by varying the hyperparameters, and the cost depends on the single instance's energy consumption for each sample of the dataset, we define $\mathcal{C}$ as:

$$\mathcal{C} = E \cdot D \tag{6.2}$$

Thereby, we want to determine how to gauge E formally. Since it is a common practice to consider only the computing device's dynamic energy consumption for comparing models [Gupta et al., 2022, Wu et al., 2022], although it does not represent the total environmental cost, we assume the following candidate sources:

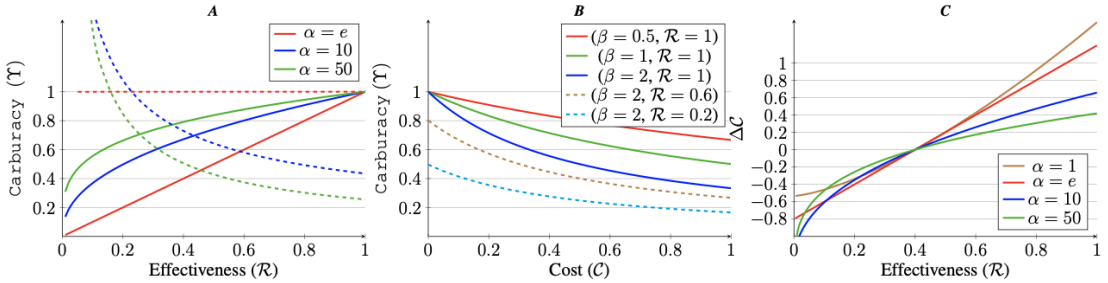


FIGURE 6.1. *A* analyzes how α impacts the CARBURACY score ($\mathcal{C} = 0$); the dashed lines are the derivatives. *B* shows different behaviors by varying β and $\mathcal{R}$ ($\alpha = 10$). *C* reports the relation between $\Delta\mathcal{C}$, the difference of carbon emissions, and the effectiveness of keeping CARBURACY stable using different α values, starting from a model with $\mathcal{R} = 0.4$ and $\Upsilon \cdot \beta = 0.5$.

- **Hardware cost**: the monetary cost of acquiring the required hardware.
- **CO_2 emissions and energy cost**: the CO_2 produced or the electricity used.
- **GPU memory usage**: the GB of GPU memory needed.

Since energy and CO_2 emissions are proportional, according to the provider policy on renewables, we only employ the latter to estimate the cost. We also neglet (1) the hardware buying cost because it is a one-time purchase strictly related to market costs and (2) the GPU memory occupation because it is unrelated to the carbon footprint, but we still report it for each experiment. Therefore, we define E as:

$$E = \textsc{Carbon}(\mathcal{M}(x)) \tag{6.3}$$

where CARBON is the kg of CO_2 produced by a model $\mathcal{M}$ to process an instance x. More precisely, CARBON is calculated by multiplying the power consumed by the computational infrastructure (quantified as kilowatt-hours) and the carbon intensity of the electricity consumed for computation (quantified as kg of CO_2 per kilowatt-hour of electricity).

6.1.3.3 Carburacy

We present CARBURACY (Υ henceforth),[2] the first carbon-aware accuracy measure, which represents the trade-off between the model effectiveness ($\mathcal{R}$) and cost ($\mathcal{C}$), as follows:

$$\Upsilon = \frac{e^{\log_\alpha \mathcal{R}}}{1 + \mathcal{C} \cdot \beta} \tag{6.4}$$

where $\alpha \geq e$ and $\beta > 0$ are hyperparameters to balance the effectiveness and the cost, respectively. Specifically, we adopt $e^{\log_\alpha \mathcal{R}}$ to simulate a non-linear trend of the

[2]We assign the upsilon Greek letter (Υ) to CARBURACY to promote the green AI concept since it resembles a tree.

effectiveness. As reported in Figure 6.1-A, the function curvature is regulated by α, exhibiting a linear behavior for $\alpha = e$. Under this setting, α rewards the effectiveness gains that are mathematically more significant. For instance, the performance improvement $\{0.05 \rightarrow 0.10\}$ is mathematically more significant than $\{0.70 \rightarrow 0.75\}$ because the first doubles the effectiveness (+100%) while the second is proportionally smaller (+7.14%). Thus, the greater the α value, the greater the cost allowed for boosting lower performance (Figure 6.1-A quantifies this reward by reporting the function's first derivative). Conversely, β weights the model cost (Figure 6.1-B), assigning more or less importance to the carbon footprint over performance.

We define $\Delta\mathcal{C}$ as the emissions produced by improving the effectiveness without decreasing CARBURACY:

$$\Delta\mathcal{C} = \frac{e^{\log_\alpha \mathcal{R}_n} - e^{\log_\alpha \mathcal{R}_o}}{\Upsilon \cdot \beta} \tag{6.5}$$

where $\mathcal{R}_n$ and $\mathcal{R}_o$ are the new and old effectiveness scores, respectively. As an example, given a model with $\mathcal{R} = 0.4$, and set $\beta = 1$ and $\Upsilon = 0.5$, Figure 6.1-C shows the $\Delta\mathcal{C}$ value needed to keep stable Υ while decreasing/increasing $\mathcal{R}$ (several trends are shown based on different α values).

The decoding strategy at inference time plays a critical role in $\mathcal{R}$, but its cost is not yet considered in Equation 6.4. For this reason, we measure CARBURACY at training and inference time, also providing their harmonic mean:

$$\Upsilon_t = \frac{e^{\log_\alpha \mathcal{R}}}{1 + \mathcal{C}_t \cdot \beta_t} \qquad \Upsilon_i = \frac{e^{\log_\alpha \mathcal{R}}}{1 + \mathcal{C}_i \cdot \beta_i}$$
$$\Upsilon_m = 2 \cdot \frac{\Upsilon_t \cdot \Upsilon_i}{\Upsilon_t + \Upsilon_i} \tag{6.6}$$

where $\mathcal{C}_t, \beta_t$ and $\mathcal{C}_i, \beta_i$ are the cost and its modulator at training and inference time, respectively. In particular, while $\mathcal{C}_t$ represents the CO_2 emissions produced to execute the training phase with n samples, $\mathcal{C}_i$ considers only the cost of processing a single test instance. Indeed, datasets have test sets of a different number of samples which, unlike training, do not contribute to model effectiveness (i.e., a test set of $10,000$ instances is more expensive to process than a test set of 100 examples, but model effectiveness does not change). In addition, since the model is trained once and applied on demand, the inference cost, which is negligible w.r.t. the training, should be evaluated for a single instance.

In this study, we empirically set $\alpha = 10$, $\beta_t = 1$, and $\beta_i = 100$, but they can be tuned differently according to what should be prioritized (i.e., model effectiveness or eco-sustainability). In detail, we choose $\alpha = 10$ because it is a good trade-off between a flat trend and a too-steep curve (Figure 6.1-A), $\beta_t = 1$ to represent the unit cost, and $\beta_i = 100$ to have the training and inference cost on similar scales.

Domain $\{\, \Upsilon \in \mathbb{R} \mid 0 < \Upsilon \leq 1 \,\}$. We obtain that using the exponential and letting $\mathcal{R} \in [0, 1]$: for $\mathcal{R} = 0$, $\log_\alpha 0 \rightarrow -\infty$ and $e^{-\infty} \rightarrow 0^+$; for $\mathcal{R} = 1$, $\log_\alpha 1 = 0$ and $e^0 = 1$.

TABLE 6.2. Statistics of the datasets used as testbeds. The number of words are averaged across all instances.

Dataset	Domain	# Docs	Source *# words*	Target *# words*
PUBMED	Biomedical	133,215	3224.4	214.4
ARXIV	Scientific	215,913	6913.8	292.8
GOVREPORT	Legal	19,466	9409.4	553.4

6.1.4 Experiments

6.1.4.1 Datasets

We contemplate the following public datasets of different domains and text sizes as evaluation benchmarks (Table 6.2).[3]

- **PubMed** [Cohan et al., 2018] consists of 133K biomedical papers from PubMed.
- **arXiv** [Cohan et al., 2018] comprises 216K scientific papers from arXiv.org.
- **GovReport** [Huang et al., 2021] includes 19K U.S. government reports.

6.1.4.2 Models

We compare the results of state-of-the-art quadratic and linear transformer-based models with a base and a large size, benchmarking 8 models overall.[4]

- **Bart** [Lewis et al., 2020a] is a transformer with a quadratic memory complexity in the input size limited to process sequences up to 1024 tokens in length due to the positional embedding mechanism.[5] We use the official `BART-base` and `BART-large` checkpoints.
- **T5** [Raffel et al., 2020] is a quadratic transformer without an input size limit, thanks to the relative embedding mechanism. We use the official 1.1 version of `T5-base` and `T5-large` pre-trained checkpoints.
- **Led** [Beltagy et al., 2020] is a transformer with a linear memory and time complexity in the input size, thanks to sparse attention. We use the official `LED-base` and `LED-large` pre-trained checkpoints.
- **LongT5** [Guo et al., 2022a] is a linear transformer with sparse attention. We use the official `LONGT5-base` and `LONGT5-large` pre-trained checkpoints with the Transient-Global attention mechanism.

[3] All datasets are publicly available in Hugging Face: https://huggingface.co/datasets. We use the "*scientific_papers*" dataset for ARXIV and PUBMED and "*launch/gov_report*" for GOVREPORT.

[4] All pre-trained model checkpoints are publicly available in Hugging Face: https://huggingface.co/models

[5] More tokens could be fed into BART by training larger positional embeddings from scratch. However, the model is not pre-trained for such embeddings, resulting in unfair comparisons.

6.1.4.3 Experiment Setup

To compare state-of-the-art models in low-resource regimes, we simulate two real-world scenarios.

- **Limited GPU memory**: we benchmark models and their GPU usage by truncating input texts into different sizes, simulating a low-resource scenario of GPU memory in which the processing of long sequences throws "out of memory" exceptions (we apply input truncation because it is the general approach used when the input is longer than the GPU memory). Technically, we used the following sizes: 512, 1024, 2048, 4096, 8192, 16384.
- **Limited number of training instances**: we analyzed models' few-shot learning ability by fine-tuning them with a different number of training samples, simulating a low-resource scenario of labeled data. Concretely, we used the following instances: 1, 10, 100, 1000, 10000.

6.1.4.4 Implementation Details

We describe the implementation details of our experiments performed with PyTorch on a workstation using a single GPU NVIDIA RTX 3090 of 24 GB memory, 64 GB of RAM, and an Intel® Core™ i9-10900X1080 CPU @ 3.70GHz.

- **Training**: we use the first n samples of the training sets without shuffling, following the same approach in [Chen and Shuai, 2021]. We train all models for 3 epochs, saving the model checkpoint that performed best on the validation sets. We apply gradient checkpointing to save memory, used the Adam optimizer, set the learning rate to $5e$-5, and set the seed to 42 for reproducibility.
- **Inference**: we evaluate all models on the first 100 instances of the test sets to save time. We use the beam search as the decoding strategy, setting the beam width to 2. We use an n-grams penalty of 3 for PubMed and arXiv and 5 for GovReport.[6] We set the following output lengths (min-max) based on statistics in Table 6.2: PubMed (100-300), arXiv (150-350), GovReport (500-1000). At the end of each training epoch, we use the same evaluation settings to monitor the performance on the first 10 samples of the validation sets (to simulate further a low-resource scenario), saving the checkpoint if $\mathcal{R}$ is higher than the performance of the previous epoch.

6.1.4.5 Metrics

We evaluated the models under several facets:

[6]The penalty ensures that no n-gram appears twice by manually setting the probability of the following words that could create an already seen n-gram to 0.

- **Effectiveness**: we used Equation 6.1 to assess the effectiveness, considering the F1 scores for r_1, r_2, r_L. In detail, we split the summaries into sentences to compute the r_L score.[7]

- **Cost**: we used Equation 6.2 to compute the models' cost. In particular, we leveraged the software CodeCarbon[8] to estimate E, namely the amount of CO_2 emissions produced by our infrastructure resources used to execute the model training and inference. Concretely, CodeCarbon monitors the CPU, TDP, and GPU energy consumption and converts the energy into CO_2 produced according to predefined conversion policy rules based on the local energy mix, defined in Lottick et al. [2019]. Despite the importance of a complete vision of the overall footprint, we omit the cost of the pre-training phase of models for several reasons: (1) Pre-training is performed once and allows models to be fine-tuned in a wide range of downstream tasks by transferring and specializing knowledge. (2) We aim to compare multiple pre-trained models in the long document summarization task. (3) The footprint of the model pre-training is on higher scales than fine-tuning, making the latter irrelevant if not applying an empirical normalization. (4) The pre-training footprint is unessential at the business level because the application costs are related to energy consumption usage. (5) The CO_2 emissions for pre-training such models, required for the overall footprint, are generally not publicly released, making the re-pre-training from scratch the only option.

- **Efficiency**: we used our CARBURACY measure (Equation 6.6) to assess the models' efficiency. Thus, we refer to efficiency as the trade-off between effectiveness and costs.

- **Occupation**: we report the GB of GPU memory used at training time to determine the models' GPU occupation.

6.1.5 Results and Discussion

6.1.5.1 Analysis on the Number of Training Samples

Figure 6.2 reports the CARBURACY scores on GOVREPORT by feed models several input sizes and training instances (the GPU memory occupation at training time is shown in Figure 6.3). The trend of CARBURACY is not linear but follows a curve with a peak around 100 training documents (Table 6.5 reports some values for greater understanding). Similar behavior can be seen from almost all models tested on different settings and datasets (we show these results in the Appendix[9] for space reasons). This trend is due to the few-shot learning capabilities of language models, which achieve good results with

[7] We used the `ROUGE` measure provided by NLG Metricverse [Frisoni et al., 2022a], using "*rougeLsum*" as r_L.

[8] https://codecarbon.io/

[9] Appendix is at https://github.com/disi-unibo-nlp/carburacy

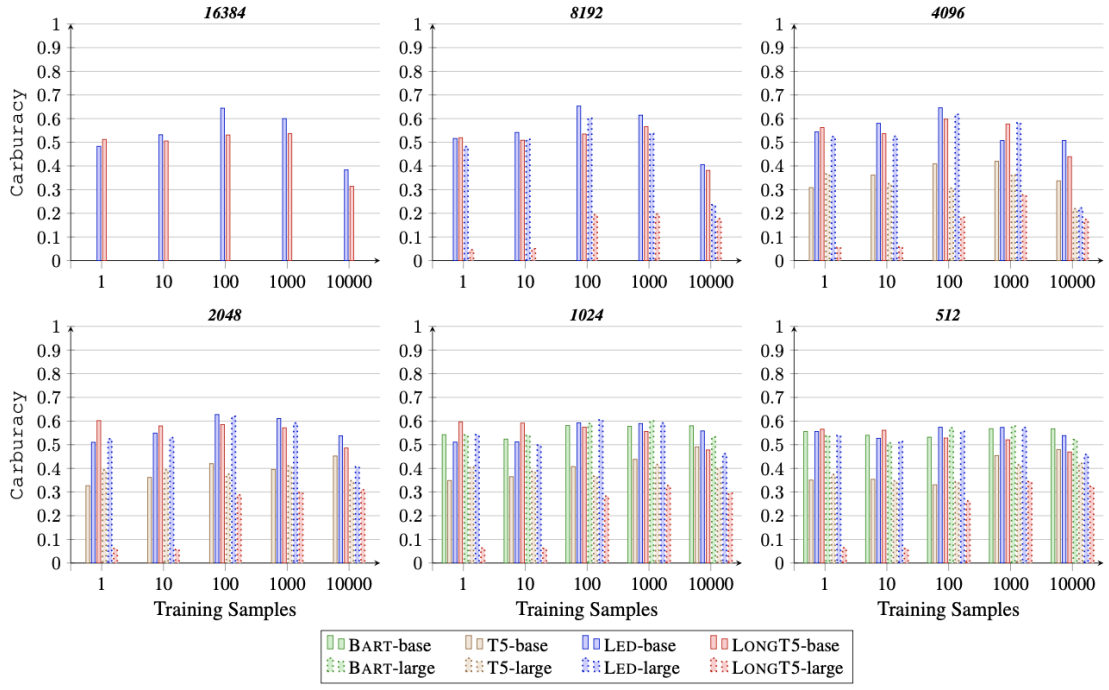


FIGURE 6.2. The CARBURACY scores on GOVREPORT by varying the input size (reported on top) and the training samples.

meager costs. Indeed, the better the few-shot learning skills, the more eco-sustainable the models because they can achieve good effectiveness with minimal CO_2 emissions. With more training samples, the extra cost does not justify the slight improvement in effectiveness, thus leading to a drop in CARBURACY. Large models, which contain more parameters, tend to obtain a similar CARBURACY score to the base models, even if they produce significantly more CO_2; but, they are inefficient for long training sessions because they emit more CO_2 without significantly improving effectiveness. Indeed, with $10,000$ samples, base models achieve a better CARBURACY score than their large versions.

6.1.5.2 Analysis on the Input Size

Figure 6.2 shows that enlarging the training input tokens results in effectiveness improvements more significant than the carbon emissions. Because of the nature of the task, which is long document summarization, extending the input tokens means giving the model more information. Thus, the same model in the same settings but with more input tokens generates a higher-quality summary. Conversely, a large number of input tokens makes the model consume more energy and GPU memory; however, the effectiveness rises significantly, increasing the overall CARBURACY. The CARBURACY score for extensive input lengths is, on average, better than for shorter ones. This trade-off makes linear transformers more eco-sustainable in long document summarization than

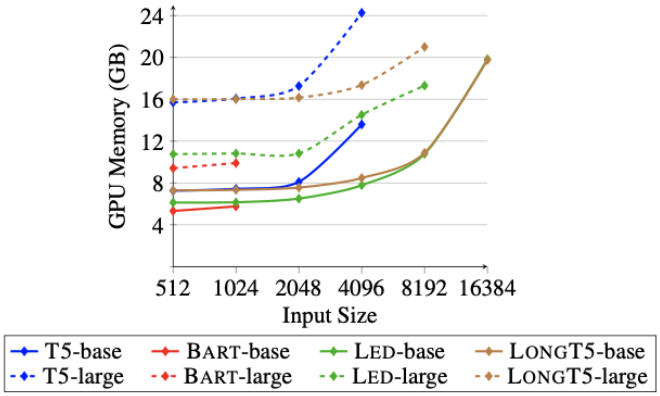


FIGURE 6.3. The GPU memory occupation on GOVREPORT by varying the input size at training time.

models limited to processing few tokens, such as BART.

6.1.5.3 Analysis on the Training Batch Size

We study whether training with different batch sizes improves CARBURACY. Specifically, Figure 6.4-B shows the results by setting the following batch sizes: 1, 2, 4, 8, 16. We notice that enlarging the batch size reduces CARBURACY slightly. This behavior is due to a drop in effectiveness while the cost remains stable. Indeed, the model learns during the backpropagation call while it adjusts the model weights to minimize the loss. Enlarging the batch size and keeping the number of training samples fixed leads to minor training steps with fewer backpropagation calls, resulting in a less-trained model. As is known, a large batch size is helpful to avoid overfitting and enhance generalization and optimal conversion, which are aspects concerning long training sessions. In our case, the training phase is relatively short, so the batch size keeps only the drawback of reducing the number of backpropagation calls, worsening the CARBURACY score of models. The only exception among all models is LONGT5-base, where CARBURACY tends to improve.

6.1.5.4 Analysis on the Decoding Strategy

The decoding strategies used to generate the final summary are fundamental for the model's effectiveness [Meister et al., 2022]. Therefore, we investigate two crucial components in the decoding phase: the input size and the number of beams in the beam search decoding method. As reported in Table 6.3 and Table 6.4, the CO_2 emissions during the test phase are significantly larger than in training, even when the latter uses more samples. Indeed, the model's encoder and decoder are only called once at training time for each input example. On the contrary, the decoder is called $j \leq k \leq z$ times during inference, with j and z equal to the min and max number of target tokens to generate, respectively. Thus, investigating the CO_2 footprint of decoding techniques is crucial, particularly for business applications.

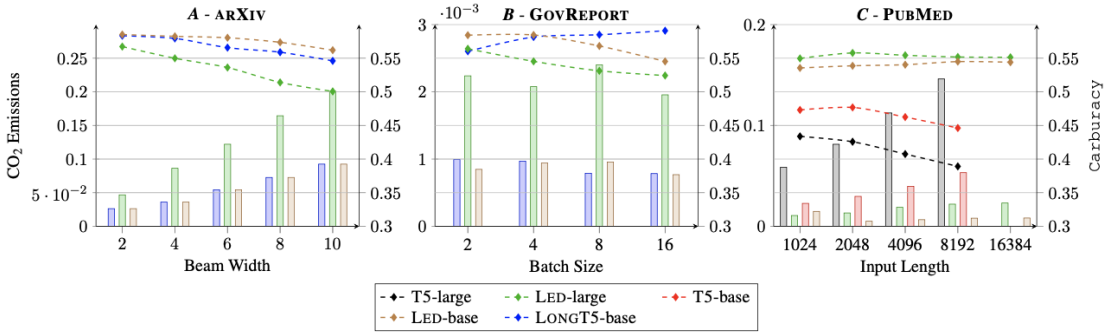


FIGURE 6.4. The graphs show the CO_2 (the bars) and CARBURACY (dashed lines) for different models. *A* reports the decoding with different beam widths. *B* shows the training with different batch sizes. *C* reports the inference with different input lengths.

TABLE 6.3. The results of the top-3 efficient base (B) and large (L) models for each dataset. Best scores are bolded for each dataset.

		Hyperparameters		*Effectiveness*		*Costs*		*Efficiency*
Dataset	**Model**	**Input Size**	**Training Samples**	**ROUGE (R1 / R2 / RL)**	$\mathcal{R}$	**kg CO_2 ($\mathcal{C}$) Train / Test**	**Memory (MB) Train / Test**	**Carburacy (Υ_t / Υ_i / Υ_m)**
PUBMED	LONGT5-B	2048	1000	39.46/14.83/35.76	29.69	0.0237 / 0.0212	7566 / 5516	55.08/**57.77**/**56.39**
	LED-L	8192	100	37.64/14.46/33.90	28.37	0.0073 / 0.0307	19540 / 11886	**56.62**/56.14/56.38
	LED-B	4096	100	35.76/12.96/32.22	26.98	0.0023 / 0.0090	8334 / 4366	56.21/55.87/55.93
ARXIV	LONGT5-B	2048	100	41.65/14.15/37.64	30.70	0.0025 / 0.0262	7734 / 6484	**59.44**/58.34/**58.88**
	LED-L	1024	100	41.08/13.22/37.29	30.07	0.0023 / 0.0157	10746 / 5704	58.93/58.43/58.68
	LED-B	8192	100	41.49/13.52/37.55	30.39	0.0042 / 0.0187	13150 / 6164	58.87/**58.52**/58.70
GOVREPORT	LED-B	8192	100	54.33/21.49/51.48	41.51	0.0036 / 0.0801	10776 / 5833	**67.54/63.20/65.30**
	LED-L	2048	100	50.40/17.70/47.37	37.67	0.0034 / 0.1021	10748 / 6254	64.79/59.38/61.97
	LONGT5-B	4096	100	50.40/17.31/47.36	37.52	0.0042 / 0.1729	8474 / 10872	64.52/55.70/59.78

Beam Search It is a decoding strategy that produces N different summaries for each document and keeps the most probable (N is the number of beams). Despite its widespread usage in text generation, it forces the model to work with multiple copies

TABLE 6.4. The comparison of base (B) and large (L) models with similar effectiveness on the PUBMED dataset.

Model	**Size**	**Samples**	$\mathcal{R}$	$\mathcal{C}$ **Train / Test**	Υ_m
Quadratic					
T5-B	1024	1000	23.07	0.0101 / 0.0229	51.52
T5-L	512	1000	24.41	0.0185 / 0.0493	51.50
BART-B	512	1000	25.80	0.0033 / 0.0041	55.14
BART-L	512	100	24.80	0.0007 / 0.0085	54.29
Linear					
LONGT5-B	4096	100	25.64	0.0027 / 0.0359	54.18
LONGT5-L	8192	10000	25.07	0.7176 / 0.0710	25.96
LED-B	8192	10	25.88	0.0002 / 0.0096	55.32
LED-L	1024	100	25.84	0.0023 / 0.0107	55.07

TABLE 6.5. The results of LED-base with 16384 as input length on GOVREPORT by varying the number of training samples.

Training Samples	$\mathcal{R}$	$\mathcal{C}$ Train / Test	Υ_t / Υ_i / Υ_m
1	26.90	0.0001 / 0.3416	56.52/42.14/48.28
10	29.85	0.0005 / 0.2230	59.06/48.37/53.18
100	42.17	0.0058 / 0.1160	**67.55**/61.59/**64.43**
1000	42.81	0.0633 / 0.1155	58.13/62.01/60.01
10000	45.67	0.5322 / 0.1146	27.40/**63.84**/38.34

of the input, resulting in high energy consumption and GPU usage. Figure 6.4-A contains the CARBURACY score of three models using the following beam widths: 2, 4, 6, 8, 10. Results show that, for each model, CARBURACY tends to decrease even when the effectiveness improves because the energy consumed is more than the performance gain. The picture also reports a linear trend of CO_2 emissions with respect to the number of beams.

Inference Input Length In this experiment, we select two linear and quadratic models: LED-large, LED-base, T5-large, T5-base. All checkpoints are trained with 1024 as the input length on 100 training samples. We investigate different inference input lengths: 1024, 2048, 4096, 8192, 16384. Figure 6.4-C reports how CO_2 emissions change according to the inference input length. Linear models keep a stable CARBURACY while CO_2 emissions increase slightly, mitigated by a minor effectiveness improvement. Quadratic models fail, with a significant CARBURACY drop due to the linear increase of CO_2 emissions. To sum up, increasing the input length at inference time does not help any model.

6.1.5.5 Findings

Q1: which are the most efficient models? Table 6.3 reports statistics of the top-3 efficient models for each dataset, which are LED-base, LED-large, and LONGT5-base. In

TABLE 6.6. The application of CARBURACY to other NLP tasks.

Model	$\mathcal{R}$	$\mathcal{C}$ Train / Test	Υ_m
Text Classification (IMDB dataset, 2011)			
DISTILBERT-base	89.64	0.0051 / 0.0012	95.12
BERT-base	89.99	0.0113 / 0.0021	94.98
Question Answering (SQuAD dataset, 2016)			
DISTILBERT-base	66.05	0.0048 / 0.0002	83.31
BERT-base	68.59	0.0080 / 0.0003	84.54

Table 6.4, we analyze the models that achieve a similar $\mathcal{R}$ on PubMed, finding that Led-base, Led-large, and Bart-base are the three most efficient models. In general, linear models are more efficient than quadratic ones, even on short documents.

Q2: which are the most impacting factors? We found that the number of training samples and beams is directly proportional to the CO_2 emissions. A similar behavior regards input size at training and inference time for quadratic models. Differently, linear models do not suffer such drawbacks because raising the training input length leads to high effectiveness and Carburacy.

Q3: which is the most eco-sustainable pipeline? Based on our findings, we initially suggest training models with small batch sizes and raising them at the end to avoid overfitting. To reduce carbon emissions while maintaining good performance, we indicate using fewer training samples (e.g., 100) and small beam widths (e.g., 2) for ablation studies.

6.1.6 Conclusion

We presented Carburacy, the first measure to score models by considering both effectiveness and costs. We compared multiple state-of-the-art transformers in long document summarization under real-world low-resource settings of hardware and data. Our results report the more environmental-friendly models and a set of hyperparameter combinations that let models achieve high effectiveness with low energy costs. Thanks to Carburacy, the research community can move toward carbon-aware state-of-the-art competitions (Table 6.6 shows the applicability of Carburacy to several NLP tasks). Future work includes extending our measure to general text mining [Frisoni and Moro, 2020, Frisoni et al., 2020a], vision language [Moro and Salvatori, 2022, Moro et al., 2023f], non-NLP tasks [Lodi et al., 2010, Cerroni et al., 2013], and ones characterized by unbounded evaluation metrics (i.e., with a score not in $[0, 1]$), such as pre-training with language modeling, a very energy-demanding session that deserves attention from an eco-sustainable point of view.

6.2
Abstractive Summarization Through the Prism of Decoding Strategies

Under review as: *Giacomo Frisoni, Luca Ragazzi, David Cohen, Gianluca Moro, Antonella Carbonaro, and Claudio Sartori. Abstractive summarization through the* Prism *of decoding strategies, 2024. Submitted to Computational Linguistics*

Abstract

In the realm of natural language generation, abstractive summarization (AS) is at the center of an unparalleled evolution driven by transformer-based language models (LMs). However, the significance of decoding strategies is often neglected, despite their influence on the generated summaries. Given the abundance of token selection heuristics and their accompanying hyperparameters, the community needs directions to steer well-founded decisions based on the task and the target metrics at hand. To fill this gap, we comparatively assess the effectiveness and efficiency of decoding-time techniques for short, long, and multi-document AS. We explore more than 2500 combinations of 3 widely used million-scale autoregressive encoder-decoder models, 6 datasets, and 9 decoding settings. Our findings shed light on the field, demonstrating that optimized decoding choices can yield substantial performance enhancements. In addition to human evaluation, we quantitatively measure effects using 10 automatic metrics, including dimensions such as semantic similarity, factuality, compression, redundancy, and carbon footprint. We introduce Prism, a pioneering dataset that pairs AS gold input-output examples with LM predictions under a wide array of decoding options.

6.2.1 Introduction

Abstractive summarization (AS) is one of the most emblematic and challenging tasks of natural language generation (NLG), aimed at condensing and rephrasing the main gist of textual documents [Sharma and Sharma, 2023]. With the advent of transformer-based solutions, autoregressive language models (LMs) have repeatedly demonstrated their prowess in generating human-like summaries [Zhang et al., 2022a]. In this red-hot research area, the AS process is typically broken down into two macro-steps: (i) training a neural network to estimate the next-token probability distributions given the input and previously predicted output tokens, (ii) applying an out-of-the-model decoding strategy to control how tokens are selected and strung together at inference time. Drawing parallelism from optics, decoding methods act like a prism: depending on how they are built and tuned with hyperparameters, they reflect model probabilities in different artificial summaries (Figure 6.5). Therefore, decoding strategies are considered one of the most significant determinants of AS output quality, also responsible for linguistic

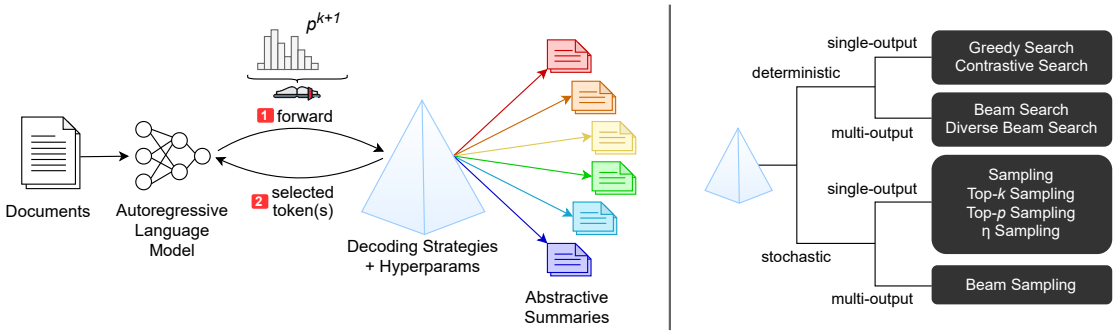


FIGURE 6.5. Conceptual division between modeling and decoding in the neural abstractive summarization pipeline (left). Decoding choices determine predicted summaries. The taxonomy (right) shows the assessed decoding strategies in this work; output n-arity refers to a single generation process.

properties, prediction n-arity, reproducibility, extrinsic hallucinations [van der Poel et al., 2022], and low information coverage [Meister et al., 2023].

Lamentably, up-to-now AS contributions lean mainly on the conservative use of default decoding settings inherited from existing tools or previous works [Shen et al., 2022]. Sometimes, the choices of the decoding algorithm are presented without much discussion [Guo et al., 2022a, Zhang et al., 2022d] or are completely omitted [González et al., 2022]. The absence of systematic practice in rigorously scrutinizing the impact of decoding raises a natural fear of its underestimation [Gong and Yao, 2023, Ji et al., 2023], fueled, among other things, by the increasing number of heuristics and the complexity of text evaluation [Frisoni et al., 2022a]. Thus, researchers urgently demand the release of comprehensive studies to shed light on best decoding practices [Zarrieß et al., 2021]. Meister et al. [2022] have recently demonstrated that decoding methods exhibit task-dependent variations, revealing that broad assertions in favor of one approach over another could lack solid grounding. However, an in-depth examination focused solely on AS is still pending.

Contributions Our paper fills this lacuna, providing the first side-by-side investigation into the effect of decoding strategies for AS, encompassing short, long, and multi-document settings (see Appendix 6.2.A for a conceptual preamble). We extensively evaluate 9 well-founded decoding heuristics across 3 state-of-the-art representative encoder-decoder models and 6 widely used datasets from different domains, exploring a broad spectrum of hyperparameters. In addition to human assessment, we put into play 10 distinct automatic evaluation metrics to scan the relationship between decoding and predicted summaries on several quality axes, including naturalness and factuality. Moreover, we judge efficiency by monitoring the carbon footprint and inference time. On balance, this work provides a blueprint for the profitable use of decoding algorithms and helps AS practitioners make confident choices that suit their needs. Our computational dedication shines through the genesis of PRISM, an innovative dataset where gold AS input-output examples are accompanied by a panoply of LM predictions and their

decoding metadata. PRISM unleashes novel analysis possibilities, which we posit could assist the community in refining NLG metrics, devising novel decoding strategies, and approximating their inherently non-differentiable behavior.

6.2.2 Related Work

Abstractive Summarization Transformers have assembled a fertile research ground for AS [Kalyan et al., 2021], with efforts to design new large-scale architectures [Guo et al., 2022a, Chowdhery et al., 2023], attention mechanisms [Huang et al., 2021, Phang et al., 2023], and pretraining objectives [Wan and Bansal, 2022, He et al., 2023]. Starting in 2019, AS supports a continuous flow of 140+ yearly publications (Figure 6.6).[10] Many of these works show interest in decoding strategies, referring to them in the title, abstract, or keywords (cf. the blue line in Figure 6.6). Cross-cutting success has been achieved even in low-resource [Moro and Ragazzi, 2022, 2023, Moro et al., 2023a,c,d] and multi-document [Moro et al., 2022a, 2023e] scenarios. The latest trends revolve around the capture of structural properties for neural document modeling [Cao and Wang, 2022] or knowledge injection [Frisoni et al., 2022b, 2023], but do not eliminate the need to choose a suitable heuristic. Reinforcement learning sequence-level rewards for direct optimization of NLG metrics have emerged as an alternative to token-level training signals [Frisoni et al., 2023, Ramamurthy et al., 2023]. In contrast, this work targets popular million-scale encoder-decoder models trained under maximum likelihood regimes.[11]

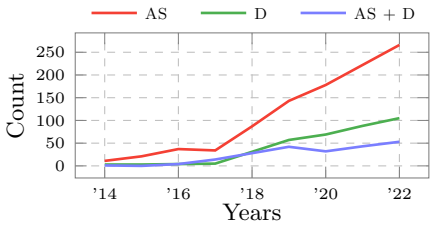


FIGURE 6.6. Annual rate SCOPUS comparison between conference papers on AS, decoding (D), and their intersection. See Appendix 6.2.B for exact queries.

Decoding Studies Prior work focused on improving the comprehension of decoding strategies, providing general overviews [Zarrieß et al., 2021], analyzes of human production variability [Giulianelli et al., 2023], or evaluations related to open-ended generation with incomplete optimization processes [Holtzman et al., 2020, Das and Balke, 2022, Su et al., 2022]. Studies on AS are often superficial (e.g., only one dataset in single-document settings [Meister et al., 2022]) or limited to the development of new heuristics without systematic comparisons [Han et al., 2019]. To our knowledge, we are the first to (i) thoroughly examine the decoding results for AS in its various forms w.r.t. input length and n-arity (i.e., single and multi-document), (ii) present findings and recommendations based on a literature-supported grid search tuning for decoding hyperparameter

[10] We direct the interested reader to Syed et al. [2021] and Sharma and Sharma [2023] for a complete overview.

[11] After a wide literature analysis, we prioritized the presently dominant LMs for AS. However, we recognize the lack of exploration in billion-scale networks (15.8% of works since 2020; details are in Appendices 6.2.B and 6.2.M). We plan to analyze large LMs, centered on decoder-only architectures and prompting strategies, in future works.

spaces, and (iii) release a pioneering decoding-oriented dataset for AS that opens new research directions.

6.2.3 Method

6.2.3.1 Probabilistic Summary Generation

We consider autoregressive encoder-decoder models, with trainable weights θ, that define the conditional probability $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ of a summary $\boldsymbol{y} = \{y_1, \ldots, y_{|\boldsymbol{y}|}\}$ given a variable-length input sequence $\boldsymbol{x}$. Depending on the AS setting, the tokens in $\boldsymbol{x}$ originate from one or more source documents, while the tokens in $\boldsymbol{y}$ are drawn from a finite vocabulary $\mathcal{V}$. The output space of hypothetical summaries is $\mathcal{Y} \coloneqq \{\text{BOS} \circ \boldsymbol{v} \circ \text{EOS} \mid \boldsymbol{v} \in \mathcal{V}^*\}$, where $\circ$ denotes string concatenation and $\mathcal{V}^*$ Kleene closure of $\mathcal{V}$; valid outputs are enclosed by special tokens "begin-of-sequence" and "end-of-sequence." The models follow a local normalization scheme, factorizing the probability of $\boldsymbol{y}$, as follows:

$$p_\theta(\boldsymbol{y}|\boldsymbol{x}) = \prod_{t=1}^{|\boldsymbol{y}|} p_\theta(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}) \tag{6.7}$$

where each $p_\theta(\cdot|\boldsymbol{x}, \boldsymbol{y}_{<t})$ is a distribution over $\bar{\mathcal{V}} \coloneqq \mathcal{V} \cup \{\text{EOS}\}$ and $\boldsymbol{y}_0 \coloneqq \text{BOS}$. Commonly, weights θ are learned by minimizing cross-entropy loss $\mathcal{L}(\theta, \mathcal{C})$ between predicted tokens and ground truth ones in a training corpus $\mathcal{C}$ (negative log-likelihood under p). To penalize overconfident output and combat overfitting, $\mathcal{L}$ can be enhanced by label smoothing, discounting a certain probability mass of the true token, and redistributing it uniformly across all other tokens [Gao et al., 2020]. The purpose of decoding is to find the most probable summary among all candidate hypotheses, i.e., $\boldsymbol{y}^* = \text{argmax}_{\boldsymbol{y} \in \mathcal{Y}}\, p_\theta(\boldsymbol{y}|\boldsymbol{x})$. This optimization problem is known as the maximum a posteriori. Since the number of possible summaries in the symbolic space increases as $|\mathcal{V}|^{|\boldsymbol{y}|}$, the exact search is NP-hard. Furthermore, even if an exact solution were tractable, it would be far from high quality text [Eikema and Aziz, 2020]. Thus, decoding is exclusively approximated with heuristic methods.

6.2.3.2 Decoding Strategies

We test the full inventory of heuristics supported by HuggingFace as of April 2023,[12] except those aimed at forcing or excluding the generation of specific tokens (out of our AS scope). Such algorithms can be set up to satisfy different NLG needs and fall into two predominant categories: *deterministic* and *stochastic*. The first optimizes for summary continuations with high probabilities, whereas the second puts randomness in place. Table 6.7 shows our decoding landscape. Note that all algorithms end when y_t or the hypotheses in $\mathcal{Y}_t$ reach EOS for some $t < max_length$. We do not consider other early stopping rules because they generally do not affect generation quality [Meister et al., 2020].

[12] https://huggingface.co/docs/transformers/internal/generation_utils

6.2.4 Experimental Setup

6.2.4.1 Experiment Goals

We aim to answer the following research questions. Q1 Is there an absolute best decoding method for AS? Q2 Are decoding methods sensitive to AS type? Q3 To what extent does proper decoding affect LM metrics? Q4 Which decoding method provides the best effectiveness–efficiency trade-off? Q5 Which hyperparameter values best suit a particular AS quality attribute?

We plan a battery of generative experiments $(S_{\mathcal{H}}, \mathcal{D}, \mathcal{M})$, where $S_{\mathcal{H}}$ denotes a decoding strategy with fixed hyperparameters, $\mathcal{D}$ is the dataset that imposes particular AS challenges and length constraints, and $\mathcal{M}$ is the autoregressive LM. In total, we executed 2656 runs (Table 6.8) for 73 GPU days. As far as we know, we drive the finest-grained hyperparameter sweep in the NLG literature. Specifically, we use grid search to simultaneously alter the generative variables of each heuristic, wisely choosing range values based on hints given in previous studies. See Appendices 6.2.D and 6.2.E for more details.

6.2.4.2 Datasets

We consider 6 noteworthy, English-language, domain-distinguished, and publicly available datasets as testbeds, 2 for each AS family. Preprocessing is carefully applied (Appendix 6.2.C).

- *Short Document Summarization (SDS).* **XSum** [Narayan et al., 2018], a dataset of BBC articles from 2010-2017, designed for highly abstractive one-sentence extreme summarization. **Cnn/Dm** [Nallapati et al., 2016], a set of articles from different news outlets accompanied by short-sentence highlights written by journalists.
- *Long Document Summarization (LDS).* **PubMed** and **arXiv** [Cohan et al., 2018], two datasets of lengthy and structured scientific papers along with their abstracts.
- *Multi-Document Summarization (MDS).* **Multi-News** [Fabbri et al., 2019], composed of news articles and human-written summaries from `newser.com`, which brings together hundreds of US and international sources. **Multi-LexSum** [Shen et al., 2022], a collection of expert-authored summaries of court documents from federal civil rights lawsuits.[13]

Data Sampling Given the huge research space for decoding and evaluation, it is impractical to work with entire datasets due to time and resource limitations. Thus, we opt for a representative subset of each dataset, leaving out the compact Multi-LexSum. We use power analysis to estimate the minimum sample size required to detect statistically significant metric effects (Appendix 6.2.C). We select a 10% dataset size by performing proportional stratified random sampling without replacement to guarantee

[13] Given the multi-granularity nature of summaries in Multi-LexSum, we contemplate the D → S task, namely synthesizing the source documents into a *short* summary.

the adequate representation of each subgroup. We stratify based on the document and summary length; tertiles are calculated to assign $\{\text{short}, \text{medium}, \text{large}\}$ classes. For MDS, we also consider the number of source documents. A compendium of the datasets is given in Appendix 6.2.C.

6.2.4.3 Models

To ensure fairness, we use 3 comparable state-of-the-art transformer-based models in their large version for which original fine-tuned weights on the datasets in Section 6.2.4.2 already exist. **Bart** [Lewis et al., 2020a] (SDS, 406M parameters) has quadratic memory complexity in input size, limited to elaborate sequences up to 1024 tokens. **Led** [Beltagy et al., 2020] (LDS, 447M) uses sparse attention to endow BART with a linear input scale, processing up to 16,384 tokens. **Primera** [Xiao et al., 2022] (MDS, 447M) adapts LED to multi-inputs through a summarization-specific pretraining objective, concatenating the sources with a special token and forming a single input of up to 4096 tokens.

6.2.4.4 Evaluation

Automatic We conjecture that AS quality estimation is similar in complexity to correctly performing the task. On the path of good practice, we take advantage of a panoply of metrics that capture separate attributes (Table 6.9). We share hyperparameters and takeaways in Appendix 6.2.E.

Human In order to better gauge the merits of the decoding, we conduct a meticulous human evaluation. Motivated by Narayan et al. [2018], Fabbri et al. [2019], and Huang et al. [2023], we use a direct comparison strategy that has been shown to be more reliable, sensitive, and less labor-intensive than rating scales. We sample 5 documents from every dataset. For each, three English-proficient AS researchers are presented with summaries inferred by 2 out of 3 sources, i.e., the top-3 decoding strategies on the dataset (optimal hyperparameter settings according to normalized average scores of effectiveness-oriented metrics, i.e., $\mathcal{R}$, BERTScore, BARTScore). We ask reviewers to select the better one w.r.t. 4 dimensions; a "tie" is declared if a judge perceives the two summaries to be of equal quality. When considering all possible combinations of summary pairs, the total number of preference labels per participant is 540. We randomize the order of pairs and summaries per example to guard the rating against being gamed. Zooming in, the rating axes are defined as follows. *Recall* considers whether the generated summary covers all target content. *Precision* checks if the generated summary covers only the target content (i.e., no redundant or superfluous information). *Faithfulness* examines whether the generated summary is factually consistent with the input document. *Fluency* dissects the grammaticality and coherence of the summaries. The final score of each decoding strategy is the percentage of times that its summaries are selected as the best, minus the percentage of times that they are not. Appendix 6.2.F illustrates our setup with human instructions.

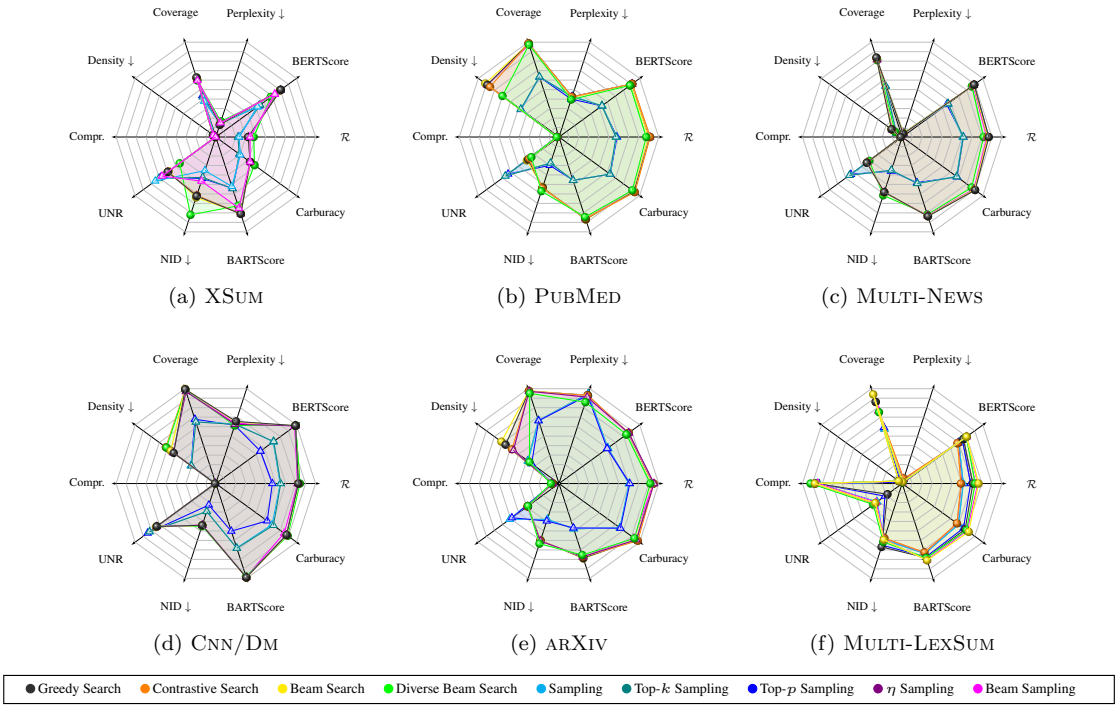


FIGURE 6.7. Metric-based comparison between artificial and gold summaries in the examined datasets. Colored areas signify the normalized average scores of different decoding strategies (● = deterministic, ▲ = stochastic). The dominant color in each dataset-specific radar denotes the best overall strategy. Succeeding Cao et al. [2022b], $[0, 1]$ rescaling is based on min-max normalization across all runs.

6.2.5 Results

The core results (averaged by the decoding method across all inference runs) are plotted in Figure 6.7; see Appendix 6.2.G for tabulated data. Please note that the enumerated quantitative findings are established on metrics that serve as proxies for the quality dimensions of interest.

6.2.5.1 Quantitative Findings

- A1 **There is no one-size-fits-all strategy for AS.** No single decoding method consistently achieves the highest quality score in all evaluation metrics. However, considering the mean average ranking in terms of $\mathcal{R}$ (the de facto standard AS metric), deterministic approaches secure the first position in every dataset. Properly calibrated stochastic methods perform exceptionally well, always covering a podium position. When adequately tuned, Diverse Beam Search demonstrates its prowess independently of the AS type. Details on metric score distributions and per-dataset rankings are provided in Appendix 6.2.H.
- A2 **Influenced by AS type, decoding strategies exhibit prevailing performance patterns across datasets.** Deterministic strategies show greater suit-

ability for AS, with Sampling, Top-k Sampling, and Top-p Sampling struggling to keep up. The consistency of these patterns is evident when comparing similar benchmarks, such as PubMed and arXiv. A deviation is observed in Multi-LexSum, which involves extremely long inputs (averaging ¿88K words, as shown in Appendix 6.2.C). In this case, truncation narrows the gap between strategies, negatively affecting Contrastive Search. The hypothesized reason is the α>0 hyperparameter, which assigns less weight to model confidence—a critical factor for handling uncertainty. Concerning AS types, Beam Search demonstrates robustness in MDS, Diverse Beam Search emerges as the preferred strategy in LDS, and Greedy Search proves particularly effective in SDS scenarios. These results raise warnings about claims made in previous publications with task-agnostic or single-AS-type settings [Su and Collier, 2023].

- A3 **Changing the decoding strategy is not just about decimals.** The decoding heuristic can substantially affect the LM scores within a given benchmark dataset. This choice can result in variations of up to 20 $\mathcal{R}$ points in PubMed, 9 BARTScore points in Cnn/Dm, and 278 Perplexity points in Multi-LexSum.
- A4 **Deterministic methods generally balance effectiveness and efficiency.** Figure 6.8 reveals that Greedy Search, Beam Search, and Beam Sampling are the fastest strategies. For Diverse Beam Search, Top-p Sampling, and η Sampling, the latency is highly variable depending on the hyperparameters. Regarding CO_2 at inference time (cf. the green line in Figure 6.7 for Carburacy), Diverse Beam Search provides the best $\mathcal{R}$–efficiency trade-off. We refer the reader to Appendix 6.2.H for Carburacy-ranked decoding strategies.
- A5 **Not all quality attributes are easy to temper at decoding time.** High beam size, high `no_repeat_ngram_size`, high temperature, and low diversity penalty promote factuality and semantic consistency. When transitioning to $\mathcal{R}$, it is strongly advised to maintain a large beam size, while the use of 0.8 temperatures in Top-k Sampling and Top-p Sampling frequently results in degeneration clusters. Interestingly, no strategy or hyperparameter can greatly favor the naturalness of text in a predictable way. However, we observe a strong positive

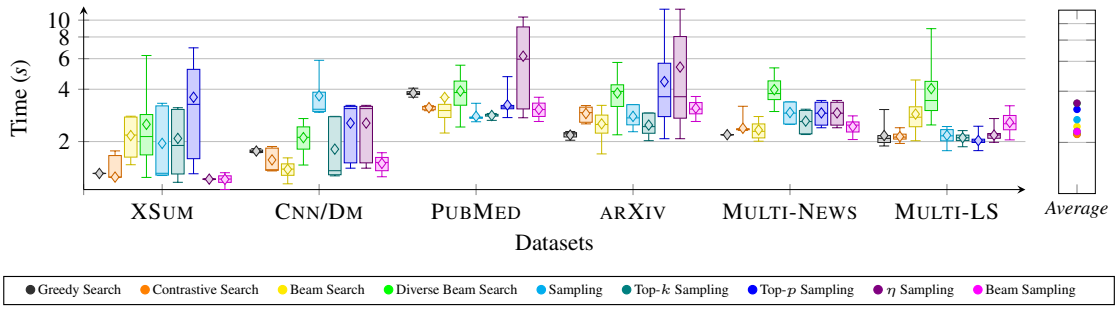


Figure 6.8. Decoding time complexity (in seconds) for summarizing a single instance of sampled dataset test sets. For each strategy, we also report the average time across all samples.

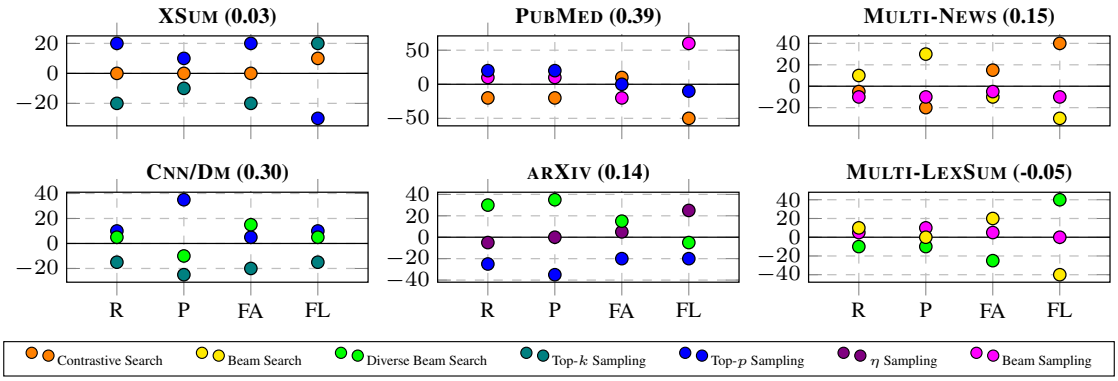


FIGURE 6.9. Win - Lose (%) human evaluation results on four quality dimensions: Recall (R), Precision (P), Faithfulness (FA), and Fluency (FL). The average Kendall's τ coefficients among all inter-annotator agreements are given in brackets.

correlation between Perplexity and Density scores (0.73 Pearson coeff.). We offer a thorough examination in Appendix 6.2.I, looking at how metrics respond to fine-grained variations in hyperparameters. Appendix 6.2.J elucidates the best hyperparameter values across all datasets, while Appendix 6.2.K shows a per-dataset evaluation.

- **Redundancy is ubiquitous, mainly in MDS.** Our scores contradict Meister et al. [2022], referring to redundancy as a rare phenomenon in AS. We pinpoint a tendency for recurring tokens as the input length increases, peaking with MULTI-LEXSUM (-60.97% UNR avg.).
- **Stochastic vs. deterministic.** Deliberate addition of randomness increases the chance of unconventional summaries and contradictions. Sampling, Top-k Sampling, and Top-p Sampling have a larger sample variance than all other strategies on five of six datasets. As expected, they have fewer repetitions (+27.63% UNR avg.) [Fan et al., 2018, Holtzman et al., 2020] but tend more to factual flaws (-40.58% BARTScore avg.) [Basu et al., 2021, Su et al., 2022]. Notably, η Sampling and Beam Sampling stand out in the stochastic sphere, suffering less from hallucinations (+40.23% BARTScore avg.). Together with deterministic methods, they produce the highest $\mathcal{R}$ (+29.65% avg.) and BERTScore (+35.34% avg.).
- **The output length matters.** Short summaries (i.e., in XSUM and CNN/DM) are less redundant than longer ones (+46.59% UNR avg.) [Xiao and Carenini, 2020].

6.2.5.2 Qualitative Findings

The annotation process took approximately 6 hours per judge. The results are presented in Figure 6.9. The average Kendall's τ of 0.16, calculated between two annotators across all pair selection results, reflects high competitiveness among the top-3 decoding strategies when correctly tuned, oftentimes leading to subjective summary preferences.

- **Top-p Sampling shines in AS with concise inputs.** While not indicated by the $\mathcal{R}$ metric on XSum, Top-p Sampling gains SDS human preference by 75%. However, its effectiveness fades as the input length increases, tipping the balance in favor of deterministic alternatives. According to this principle, Beam Search prevails in MDS.
- **Fluency negatively correlates with Recall, Precision, and Factuality**. Fluency goes opposite to Recall, Precision, and Factuality more than 66% of the time. Furthermore, Fluency preferences depend on the dataset and do not always reward stochastic strategies.

6.2.6 The Prism Dataset

Composition Building upon the experiments presented in the previous sections, we introduce Prism, a first-of-its-kind dataset that collects over 2M artificial summaries generated over a range of heterogeneous decoding settings. Prism presents an instance for each inference run, detailing all metadata (dataset, model, decoding strategy, hyperparameter values), average decoding time per instance (milliseconds), carbon emissions (kg), and metric scores. Its source files (≈10 GB) are stored in JSONL format and are publicly available for download through the HuggingFace Datasets platform.[14] For example, to access the summaries predicted by a Beam Search run, you need to install the `datasets` Python library and follow the instructions shown in Figure 6.10. For the sake of space efficiency, we separately release the gold document-summary AS pairs that each run relies on. Additional information on the project website[15] will be updated regularly to incorporate any future changes, additions, or erratum.

```python
from datasets import load_dataset
# Download PRISM locally and load it as a Dataset
prism = load_dataset("PRISM")
# The first Beam Search run
run = prism["beam_search"][0]
# The predicted summaries
run["predictions"]
```

Figure 6.10. Prism HuggingFace Dataset.

Applications The potential applications of Prism are extensive and diverse. Researchers can exploit this dataset to study new NLG metrics [Frisoni et al., 2022a]. Additionally, it provides a unique opportunity to benchmark decoding strategies against a multitude of established baselines. Beyond this, Prism offers the ability to train LMs to emulate the token choice of one or more strategies for style control [Goyal et al., 2022b] or automatic hyperparameter optimization [Chen et al., 2022b]. In fact, decoding strategies are complex algorithms that are hard to put in end-to-end networks due to their non-differentiable nature. In light of the poor attempts to design exact differentiable versions of decoding strategies (e.g., Top-k Sampling [Jang et al., 2017], Beam Search [Collobert et al., 2019]), Prism emerges as an indispensable asset for creating approximated modules.

[14] https://huggingface.co/datasets/[anonymized]/PRISM
[15] https://prism.github.io. The web page will be public at the end of the anonymization window.

6.2.7 Conclusion

The rocketing growth witnessed by transformer-based summarizers is offset by the poor control over decoding strategies, which exhibit cloudy task-specific qualities overshadowed by the continuous distribution of new models. In this paper, we demystify the role of decoding-time methods for abstractive summarization. Our full-scale study comprises comprehensive quantitative and qualitative breakdowns, covering various decoding setups, autoregressive models, datasets, and evaluation metrics. Empirical results demonstrate how generative heuristics and their hyperparameters can overturn predicted summaries, where optimal choices depend on target quality dimensions and the summarization type at hand (i.e., long, short, multi-document). Besides validating observations already made in other tasks, our findings unveil the uniqueness of abstractive summarization and the best procedures to follow depending on the case, serving as cautionary notes. Wrapping up our core findings, we furnish practitioners with a practical and easy-to-follow guideline (Appendix 6.2.L), facilitating the right selection of decoding strategies and hyperparameters tailored to specific case studies. Our breath study and the data collected unlock new research avenues, raising expectations for a future marked by increased awareness of the implications of decoding and their control.

Reproducibility Statement

To help readers reproduce our experiments, we provide rationales for our decoding hyperparameter search space in Appendix 6.2.D, listing each dataset's minimum and maximum input/output length. All models, datasets, decoding strategies, and automatic metrics explored in this study are open source; Appendix 6.2.E elaborates on implementation specifics, hardware setup, and runtimes. Since our decoding runs are performed on representative dataset samples, we also include details of the preprocessing steps and the power analysis process completed before sampling (Appendix 6.2.C). Appendix 6.2.F describes our human evaluation protocol. We plan to openly release the source codes in a dedicated GitHub repository and PRISM on HuggingFace Datasets.

Ethics Statement

We honor ICLR Code of Ethics. As we recognize that the reported principles are not exhaustive, we address the nine points explicitly mentioned in the NeurIPS 2023 Ethical Guidelines.

1. Does the data contain any personally identifiable information or sensitive personally identifiable information? Our data do not contain confidential information. All source documents are available for free inspection, uncopyrighted, and fully public.

2. Does the data contain information that could be deduced about individuals that they have not consented to share? Our data contain individual names.

Nevertheless, such details are within news reports published by authoritative sources, such as BBC and CNN, on which we rely.

3. Does the data encode, contain, or potentially exacerbate bias against people of a certain gender, race, sexuality, or who have other protected characteristics?
No.

4. Does the paper contain human subject experimentation and whether it has been reviewed and approved by a relevant oversight board?
No.

5. Does the paper rely on data that have been discredited by the creators?

No.

6. Consent to use or share the data. Explain whether you have asked the data owner's permission to use or share data and what the outcome was.
Consent is implicit for all content because the datasets used are publicly available.

7. Domain specific considerations when working with high-risk groups
Not applicable.

8. Filtering of offensive content. For instance, when collecting a dataset, how are the authors filtering offensive content such as racist language or violent imagery?
Not applicable. Our data does not involve offensive content.

9. Compliance with GDPR and other data-related regulations. For instance, if the authors collect human-derived data, what is the mechanism to guarantee individuals' right to be forgotten (removed from the dataset)?
As noted in the paper, our data are derived from multiple publicly available datasets, including news reports, scientific papers, and court lawsuits that do not comply with privacy rules.

6.2.A Abstractive Summarization Families

We define the problem of AS with the following setup. For single-source tasks (SDS and LDS), the input is a document $\mathcal{X} = \{x_1, \ldots, x_x\}$, where each $x_i \in \mathcal{X}$ is a token. For multi-source scenarios (MDS), the input is a cluster $\mathcal{C} = \{\mathcal{X}_1, \ldots, \mathcal{X}_z\}$ of documents.

- In *SDS*, $\mathcal{X}$ is a brief document, generally shorter than 1024 tokens, which is the maximum size that transformer-based LMs with quadratic complexity [Lewis et al., 2020a, Zhang et al., 2020a] can process without input truncation.

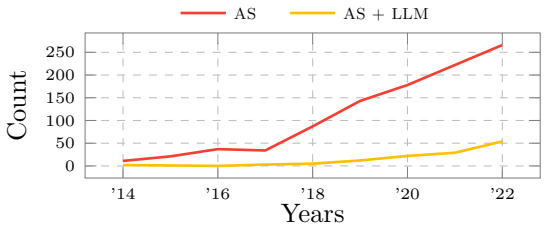


FIGURE 6.11. Annual rate SCOPUS comparison between conference papers on abstractive summarization (AS) and their intersection with large language models (LLMs).

- In *LDS*, the number of input tokens could be potentially large (e.g., $>$ 10,000). For this reason, quadratic LMs would ignore summary-worthy information and are thus replaced by efficient transformers with linear complexity that can read up to 16,384 tokens [Beltagy et al., 2020, Huang et al., 2021, Guo et al., 2022a, Phang et al., 2023].
- In *MDS*, $\mathcal{C}$ is a cluster consisting of multiple documents related to a topic (e.g., newspaper articles detailing the same recent event). Generally, $\mathcal{X}$ is assembled by concatenating the documents in $\mathcal{C}$ to form a single long textual input [DeYoung et al., 2021, Xiao et al., 2022], treating the summarization problem as in LDS.

6.2.B SCOPUS Queries

The bibliometric results reported in the main paper (Figure 6.6) are obtained by executing the following queries on the SCOPUS search engine:

- **Abstractive Summarization**
  `TITLE-ABS-KEY(abstractive AND summarization);`
- **Decoding Strategies**
  `TITLE-ABS-KEY(((text AND generation) OR nlg) AND decoding);`
- **Abstractive Summarization + Decoding Strategies**
  `TITLE-ABS-KEY(summarization AND decoding);`

We also show in Figure 6.11 the intersection between abstractive summarization and large language models (LLMs), obtained with the subsequent query:

- **Abstractive Summarization + Large Language Models**
  `TITLE-ABS-KEY(abstractive AND summarization AND ((large AND language AND (model OR models)) OR LLM)).`

We only consider conference papers on Computational Science by appending the following: `AND (LIMIT-TO(SUBJAREA, "COMP")) AND (LIMIT-TO(DOCTYPE, "cp")).`

6.2.C Dataset Preprocessing and Power Analysis

According to best practices in summarization corpora construction [Kornilova and Eidelman, 2019], our preprocessing pipeline foresees (i) lowercasing, (ii) ASCII encoding, (iii) extra space, URL, special character, and bullet point removal, (iv) HTML cleaning, (v) dash sequences deletion, (vi) newline and tabs erasure, (vii) empty sentence and non-alphabetic starter cut, (viii) between-sentence demarcation space assurance, (ix) word contraction expansion, and (x) quotes normalization.

A prudent preliminary operation is to determine the required sample size to ensure that the study has sufficient statistical power to detect meaningful metric effects across decoding runs if they exist. To this end, we perform a power analysis for the t-tests using the `statsmodels` Python library (v 0.14.0). The resulting needed sample size is 393.4. Technically, we consider the following factors:

- *Power level.* It denotes the probability of correctly detecting a true effect if it exists. Higher power indicates greater ability to detect effects. We use the commonly recommended value 0.8, which corresponds to an 80% chance of correctly detecting a true effect.
- *Significance level.* It is the threshold used to determine statistical significance. We choose 0.05, which corresponds to a 5% chance of incorrectly rejecting the null hypothesis.
- *Effect size.* It is the magnitude of the difference that we expect to observe among different decoding strategies. We employ Cohen's d, calculated by taking the difference in means between two groups and dividing it by the pooled standard deviation, i.e., $(\bar{x_1}-\bar{x_2})/s$. Following preliminary experiments, we make an educated guess about the aggregate effect size across all NLG metrics. We set $\bar{x_1} - \bar{x_2}$ at 0.01 and s at 0.05.

After preprocessing, the dataset statistics (before and after sampling) are listed in Table 6.10. The number of words is derived from `nltk.word_tokenize` [Bird, 2006].

6.2.D Decoding Hyperparameters

For *Contrastive Search*, we modulate α and k in $\{0.2, 0.4, 0.6, 0.8, 1.0\}$ and $\{20, 30, 40, 50, 60\}$, respectively. For *Beam Search*, we choose the beam size b from $[2, 10]$, a large window that allows a trade-off between quality and performance. We keep b unchanged for *Diverse Beam Search*, implementing Δ with the Hamming distance, as suggested by Vijayakumar et al. [2018], and considering a diversity penalty $\lambda \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$; the number of subgroups is set equal to b. We keep the same exploration range for b when we try out *Beam Sampling*. For *Top-k Sampling*, we pick k from $\{20, 30, 40, 50, 60\}$. For *Top-p Sampling*, supported by the results of DeLucia et al. [2020], we choose p from $\{0.4, 0.6, 0.8\}$. For η *Sampling*, we search η-cutoff over $\{0.0003, 0.0006, 0.0009, 0.002, 0.004\}$ [Hewitt et al., 2022]. For pure (ancestral), Top-k,

and Top-p Sampling, we also perform an ablation on the temperature hyperparameter, exploring $\tau \in \{0.8, 0.9, 1.0\}$, the best values according to Holtzman et al. [2020] and Pasunuru et al. [2021]. Where not specified, we maintain default hyperparameters depending on the model configuration.

Artificial summaries often suffer from high redundancy [Xiao and Carenini, 2020]. Therefore, we investigate non-negligible word-level n-grams penalties as introduced by Klein et al. [2017] and Paulus et al. [2018], precisely for AS. In a nutshell, we force the decoder to never output the same sequence of words more than once during testing by manually setting the probability of already hypothesized n-grams to 0. We extract n from $\{3, 4, 5\}$ for PubMed, arXiv, Multi-News, and Multi-LexSum, and from $\{2, 3\}$ for XSum (one-line output) and Cnn/Dm. This setting gives a minimum of 16 runs per decoding strategy.

As common in AS, we define a minimum and maximum summary length for each dataset (Table 6.11), avoiding the production of an eos token before or after a certain threshold. These constraints are determined by examining the summary sizes in the validation tests.

Decoding Runs There is a strong paucity of research on generative effects governed by the available next-token selection strategies. To our knowledge, only two systematic benchmarks focus on intertask behaviors and output diversity [Ippolito et al., 2019, Meister et al., 2022]. Most comparative studies originate primarily from papers that introduce novel decoding techniques with the intention of providing either theoretical justification or empirical evidence for their effectiveness. Although the selection of the decoding strategy is widely known to be task-dependent, the emphasis in community efforts primarily lies on open-ended generation, leaving several questions unanswered in application fields such as AS. In this area, Beam Search is often lazily treated as the de-facto standard, with limited exploration of alternative decoding strategies [Beltagy et al., 2020, Lewis et al., 2020a, Xiao et al., 2022]. Despite the collective superficiality and the latest heuristics, we want to step back and question this leap of faith. Furthermore, research efforts that already incorporate AS often draw task-level conclusions based on the analysis of a single dataset and a limited research scope, resorting to SDS for the sake of experimental efficiency [Meister et al., 2022, Su and Collier, 2023]. Fanning the flames, the efficiency data on runtime and CO_2 are rarely mentioned. Compared to previous work (Table 6.12), our task-specific investigation is offset by the high diversity of heuristics and their settings, detailing the characteristics of the SDS/LDS/MDS family. Consequently, we consider this work the most extensive decoding study, also w.r.t. the diversity of AS models evaluated. For example, previous research by Meister et al. [2022] and Su et al. [2022], Su and Collier [2023] considered exclusively one AS model each: Bart [Lewis et al., 2020a] and Gpt-2 [Radford et al., 2019b], respectively.

6.2.E Implementation Details

Hardware Configuration Inference runs are performed on a workstation that has 4 Nvidia GeForce RTX3090 GPUs with 24 GB of dedicated memory each, 64 GB VRAM,

and an Intel® Core™ i9-10900X1080 CPU @ 3.70GHz.

Models, Datasets, and Decoding Our code is founded on PyTorch 1.10.2 [Paszke et al., 2019], the HuggingFace Transformers [Wolf et al., 2020] and Datasets [Lhoest et al., 2021] libraries. Table 6.11 and Table 6.13 enumerate datastore references and model checkpoints fine-tuned on them. Sticking to Xiao et al. [2022], the input for PRIMERA is the concatenation of documents within the clusters (in the same order), where truncation is applied to each document based on the input length limit divided by the size of the cluster, thus guaranteeing the representation of all sources. All decoding runs for autoregressive summary generation are subject to the `generate()` method of HuggingFace. We set the global seed to 42 to guarantee the reproducibility of all runs of our work.

Metrics We quantify automatic metric scores using NLG-METRICVERSE [Frisoni et al., 2022a], moving to external official repositories where it is impossible. Table 6.14 lists all hyperparameters. Due to the greater correlation with human judgment, we computed BERTScore with DEBERTA-large instead of the default ROBERTA-large, as recommended by the authors from version 0.3.11. We conducted tests using SummaC [Laban et al., 2022], which is considered the main factuality metric for abstractive LDS [Koh et al., 2022]. However, its computational demands proved to be too time-consuming, leading us to prefer BARTScore.

TABLE 6.14. Hyperparameters of the NLG metrics.

Metric	Hyperparameters
ROUGE	rouge_types=["rouge1","rouge2","rougeLsum"], use_stemmer=True
BERTScore	lang="en", model_type="microsoft/deberta-large-mnli", idf=True, batch_size=32, rescale_with_baseline=True
Perplexity	model_id="gpt2", add_start_token=False
BARTScore	checkpoint="facebook/bart-large-cnn", batch_size=4
Carburacy	alpha=10, beta=100

Experiment Tracking We tracked all our runs with Weights & Biases[16] and monitor CO_2 emissions with CodeCarbon.[17] Our experiments can be publicly found on our Weights and Biases project.[18] The cumulative processing time of the GPU for generating all summaries amounts to ≈60 days. Computing all metric scores requires ≈13 days: 12 hours for $\mathcal{R}$, 18 hours for BERTScore, 3.6 hours for Perplexity, 198 hours for Coverage, Density, and Compression, 38 minutes for UNR and NID, 75 hours for BARTScore, and 10 seconds for Carburacy. Table 6.15 reports FLOPs computational performance of the decoding strategies as an additional measure of processing power and efficiency.

[16] https://wandb.ai
[17] https://github.com/mlco2/codecarbon
[18] https://wandb.ai/decoding-summ/prism

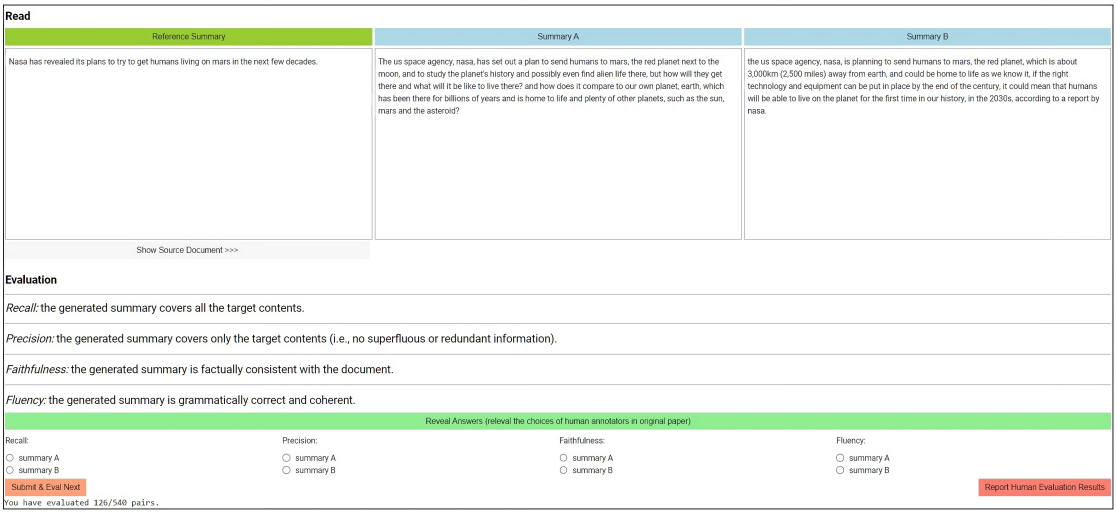


FIGURE 6.12. Screenshot of the interface used to perform the human evaluation.

6.2.F Human Evaluation

Our setup is sketched in Figure 6.12, inspired by Gu et al. [2022]. We instruct the annotators to distinguish quality dimensions by presenting the example reported in Table 6.16.

6.2.G Normalized Tabulated Scores

For better interpretability, Table 6.17 offers the tabulated version of Figure 6.7. Data are categorized by dataset, metric, and decoding strategy, with a 0-10 normalization imposed by the magnitude of values and graphical reasons. We prioritize the relative relationships among the strategies rather than the precise score values achieved by each. Please note that the exact metric scores are openly released in our PRISM dataset.

6.2.H Metric Score Distributions

We show the distributions of the metric scores by decoding strategy and dataset in Figure 6.13. Depending on the hyperparameters chosen, Sampling, Top-k Sampling, and Top-p Sampling display the highest degree of result variability. Out of the deterministic methods, Diverse Beam Search is characterized by high redundancy. MDS consistently yields a lower perplexity than the other task families, reflecting the difficulty of current models in achieving high naturalness. The extractive nature of a summary in LDS scenarios can vary greatly depending on the strategy implemented. As expected, the length of the target is strongly correlated with the $\mathcal{R}$ score, which implies a higher probability of generating n-grams of overlap with longer references, especially in LDS scenarios.

We offer per-dataset rankings on $\mathcal{R}$ (lexical overlap, Figure 6.14) and Carburacy

($\mathcal{R}$–efficiency trade-off, Figure 6.15), recognized as useful metrics for practical applications. Visualizing the decoding strategy ranks offers a complementary perspective for a comparative fine-grained evaluation alongside individual and averaged observations. We found `temperature=0.8` in Top-k Sampling and Top-p Sampling being the reason for the pronounced negative metric score clusters in XSum, Cnn/Dm, PubMed, arXiv, and Multi-News. The lowest ranks in Multi-LexSum are attributable to η Sampling and Contrastive Search.

6.2.I On Hyperparameters Effect

To encourage a conscious configuration of decoding strategies, we now consider the relation between each hyperparameter value and the primary metrics (i.e., $\mathcal{R}$, BERTScore, BARTScore, Perplexity).

Overall Unexpectedly, while the value of `no_repeat_ngram_size` (abbreviated as `no_rep`) rises, there is a decrease in $\mathcal{R}$ and BERTScore, but a simultaneous increase in factuality (BARTScore). Our conjecture is that this can be explained by the elevated abstractiveness of the datasets, a hypothesis reinforced by human evaluation. In fact, there is a tension between staying close to the source document and allowing abstractive modification. Taking a wider perspective, we find that `no_rep`, a frequently overlooked parameter, plays a substantial role in shaping the output.

Greedy Search (Table 6.20) In the case of SDS and LDS, an increase in `no_rep` is accompanied by a corresponding increase in the value of $\mathcal{R}$, except for XSum. MDS follows an inverse pattern: with the growth of `no_rep`, both $\mathcal{R}$ and BERTScore decline, while BARTScore increases. The data show a notable correlation between tasks, with only two outliers: $\mathcal{R}$ and BERTScore in XSum, and Perplexity in Multi-LexSum.

Contrastive Search (Table 6.21) It is difficult to pinpoint the selection guidelines for `top_k`, but lower values achieve better results. Decreasing `no_rep` in XSum leads to improved $\mathcal{R}$ scores. In Multi-LexSum, choosing a low `penalty_alpha` is preferable.

Beam Search (Table 6.22) When delving into the SDS scenarios, it becomes evident that factuality experiences enhancement as both `no_rep` and beam size b increase. Looking at BERTScore, we also detect a positive trend with `no_rep` and b as well, peaking at $b = 4$ and $b = 5$. On the contrary, when examining XSum, there is a declining trend in $\mathcal{R}$ relative to high values for `no_rep` and b. However, this trend is less pronounced in Cnn/Dm, which benefits from a central b and `no_rep` $= 3$. This dissimilarity can likely be ascribed to the varying length of the summaries. Specifically, since XSum is much more concise, imposing a stringent non-repetition constraint can significantly affect lexical similarity. This opposition is also apparent in terms of Perplexity. As for LDS, arXiv and PubMed experience an increase in $\mathcal{R}$ and BERTScore with high `no_rep` and low b. Again, the value of b sets a boundary on the attainable level of factuality. In the context

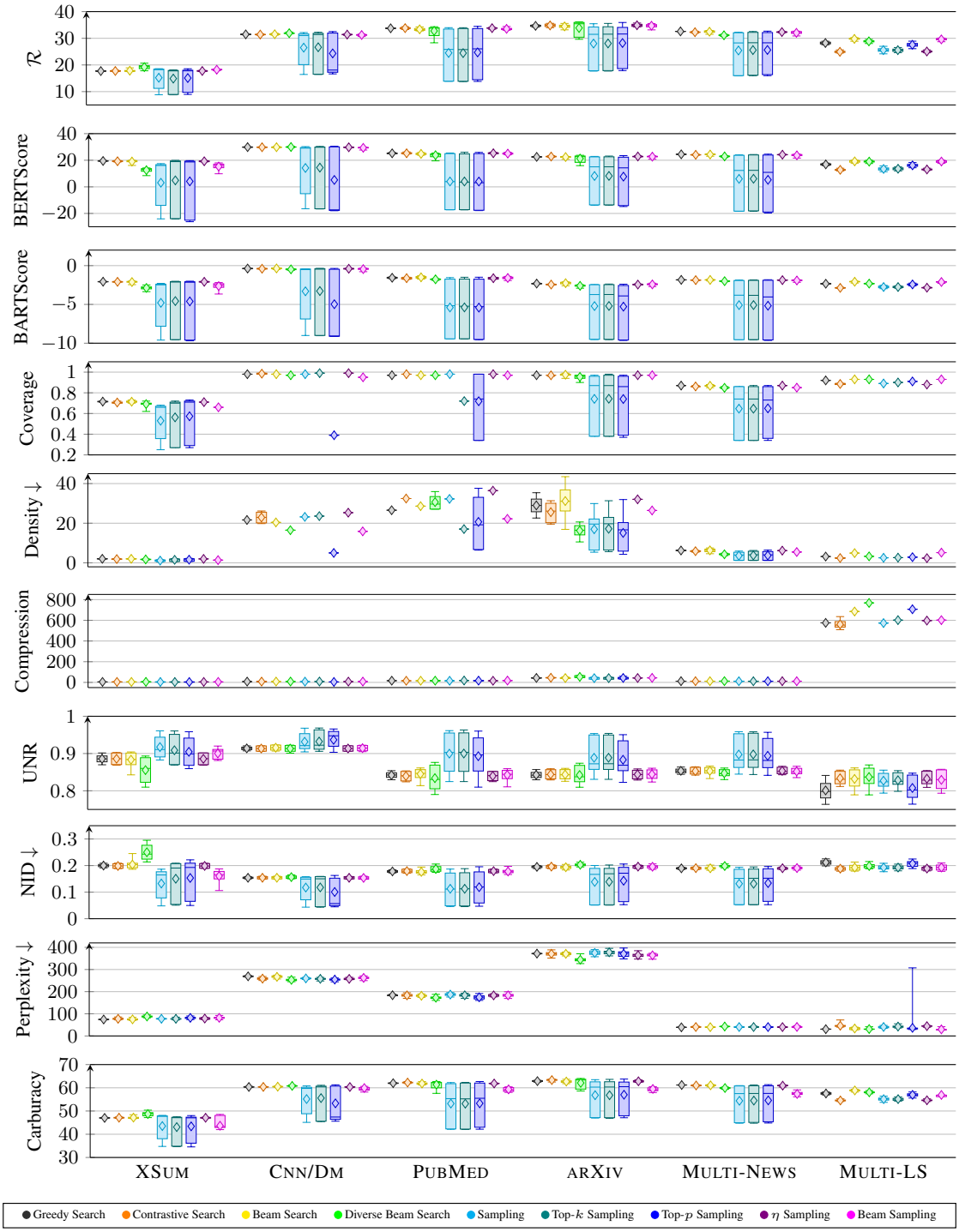


FIGURE 6.13. Graphical metric score distributions grouped by datasets and strategies.

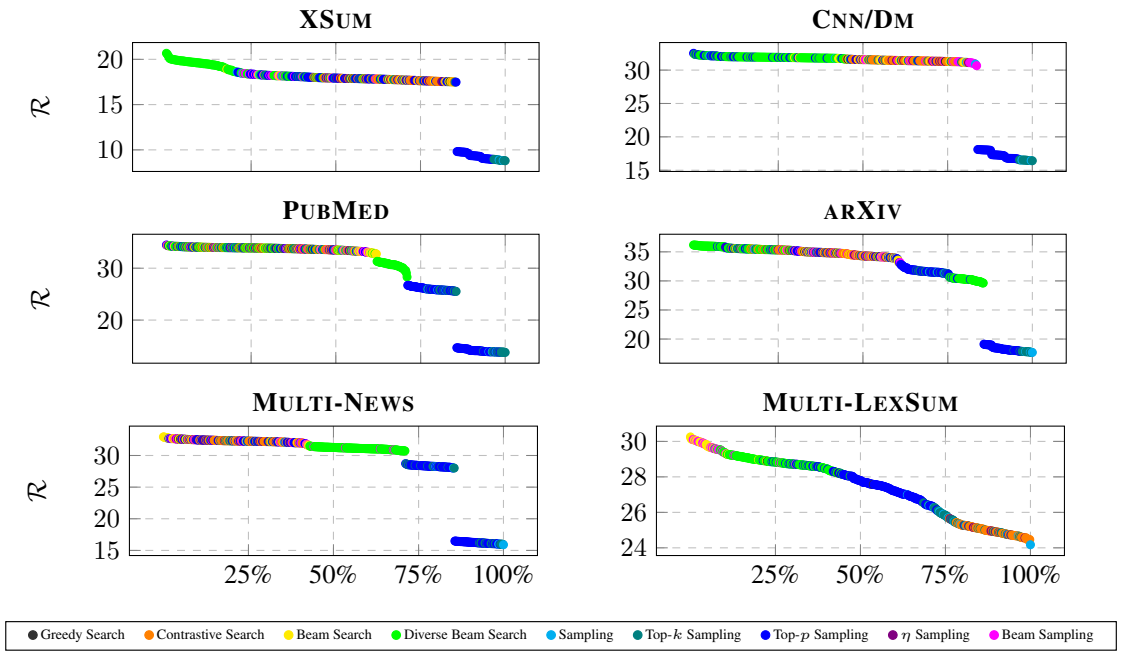


FIGURE 6.14. Per-dataset decoding strategy ranking according to $\mathcal{R}$.

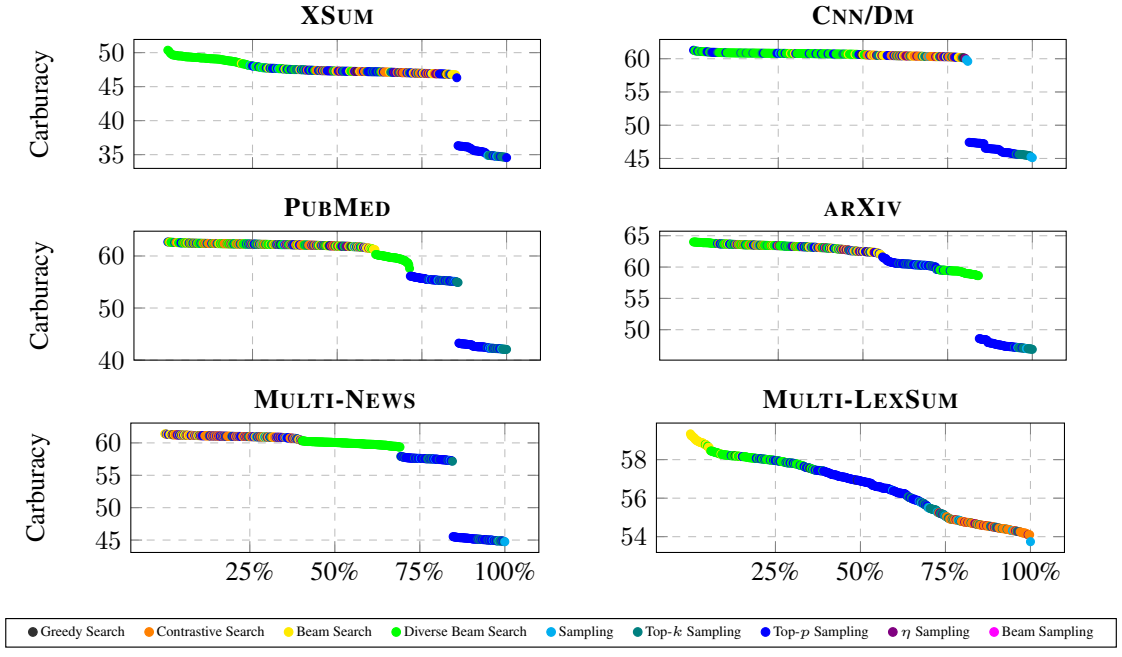


FIGURE 6.15. Per-dataset decoding strategy ranking according to Carburacy.

of MDS, there is a remarkable shift in the trajectory of $\mathcal{R}$, which increases with `no_rep` and b. BARTScore maintains the same pattern as shown in SDS and LDS. BERTScore

fluctuates more and confirms its dataset-specific conduct, but generally benefits from higher values of b, irrespective of `no_rep`.

Diverse Beam Search (Table 6.23) In SDS, artificial summaries from XSum achieve high $\mathcal{R}$ scores when there is an increase in `no_rep`, a decrease in b, and a low Λ value of `diversity_penalty`. On the other hand, better semantic alignment is achievable with low `no_rep`, high b, and low Λ. Optimally tuned hyperparameters make Diverse Beam Search superior to standard Beam Search. In LDS, the $\mathcal{R}$ trend is unchanged, and Λ has less impact on the syntactic surface. In particular, the effectiveness curve according to BERTScore changes radically, favoring high `no_rep`, low b, and low Λ. Moving to MDS, the general results of Multi-News improve with increasing b and Λ. In all task families, there is a consistent upward trend in BARTScore as b and `no_rep` values increase, except for XSum.

Sampling (Table 6.24) In SDS and MDS, the behavior is well-established, suggesting the adoption of high temperatures to maximize $\mathcal{R}$, BERTScore, and BARTScore. The only outlier is Multi-LexSum, hypothesizing that it is due to the extreme length of its inputs, which causes inaccurate summaries to be generated. Perplexity varies across datasets and shows a weak correlation with changes in hyperparameters.

Top-k Sampling (Table 6.25) and Top-p Sampling (Table 6.26) Excluding Multi-LexSum, $\mathcal{R}$, BERTScore and BARTScore improve as temperatures increase. When compared to temperature, `top-k`, `top-p`, and `no_rep` have very little effect on the metrics. The disparity between Top-k and Top-p Sampling mainly arises from Perplexity in longer inputs, favoring the latter approach.

η Sampling (Table 6.27) The behavior of η Sampling appears to be difficult to predict and control. As for BARTScore, results get better with high `no_rep`, while `cut_off` has a large impact only on Multi-LexSum. The rise in $\mathcal{R}$ often aligns with an increase in `no_rep`, while variations in `cut_off` exhibit a comparatively weaker impact.

Beam Sampling (Table 6.28) In the case of XSum, there is a conspicuous positive correlation between b and the metrics $\mathcal{R}$, BERTScore, and BARTScore (`no_rep` has a small influence). A similar pattern is observed for Cnn/Dm, where optimal performance is achieved with $b = 6$, followed by slight oscillations. In LDS, $\mathcal{R}$ and BERTScore have opposite trends in PubMed and arXiv. In MDS, $\mathcal{R}$ increases with b and `no_rep`, with few exceptions in Multi-LexSum.

6.2.J Best Hyperparameters

Table 6.18 presents the hyperparameters that lead to the best $\mathcal{R}$ score (i.e., ≥ 0.9 quantile).

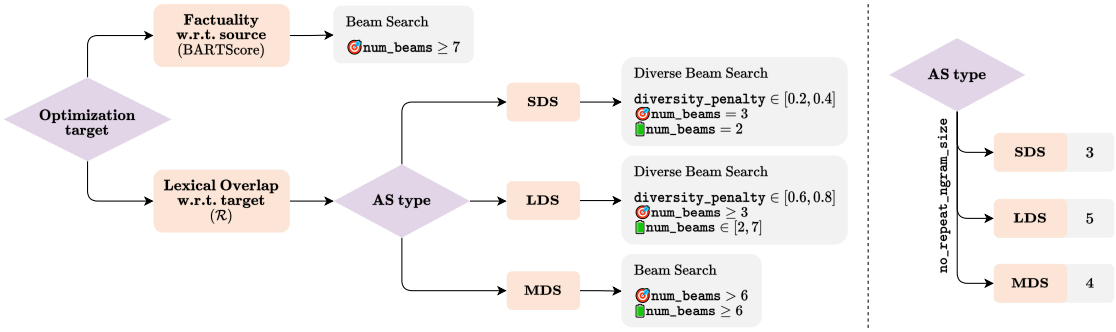


FIGURE 6.16. Visual guideline for decoding strategy selection. Optimization target = "What are you interested in?"; AS type = "What is your Abstractive Summarization scenario?". The hit symbol, when included, denotes hyperparameter value recommendations for achieving maximum effectiveness on the proxy metric. In contrast, the battery symbol signifies the optimal effectiveness-efficiency trade-off as proposed by Moro et al. [2023b].

6.2.K Best Decoding Strategies per Dataset

Table 6.19 reports the configuration of the top-3 $\mathcal{R}$ decoding strategies per evaluated dataset.

6.2.L Guideline

Based on average scores, ranking positions, and hyperparameter sensitivity, we suggest a small-size decoding space depending on the optimization target and AS type (Figure 6.16).

6.2.M Limitations and Future Directions

Additional Decoding Strategies We focused on broadly applicable decoding methods that implement the same likelihood objective as the models. However, some recent strategies add substantial further assumptions, goals, non-metric-agnostic outputs, and task-specific constraints to generation. Among these, we mention FAME [Aralikatte et al., 2021], which dynamically biases the decoder to proactively generate summary tokens that are topically similar to the input, and PINOCCHIO [King et al., 2022], which restricts Beam Search only to consider tokens that are likely to be supported by the source text. Future work should investigate their inclusion in the analysis.

Additional Models We conducted a diligent model selection process to identify representative state-of-the-art encoder-decoder solutions within each AS family (short, long, multi-document). Chosen models have gained widespread recognition and feature fine-tuned checkpoints from reputable sources (Meta and AllenAI). In the scope of this study, our emphasis was on evaluating the models most commonly used by the AS community,

rather than those that might gain prominence in the future. These models belong to a million parameter scale or hardly overcome 10B, a lower boundary often cited when discussing LLMs [Zhao et al., 2023]. Extending our study to LLMs such as GPT-3 [Brown et al., 2020] and open source alternatives would be noteworthy. However, we underline that resorting to multiple LLMs would be essential to avoid reaching decoding-level conclusions solely based on the behavior of a particular model (and not an entire spectrum of modeling techniques), an aspect frequently overlooked in previous publications [Meister et al., 2022, Su et al., 2022, Su and Collier, 2023]. Furthermore, limiting the LLM analysis to a subset of AS types would not be a sound assessment choice. Also, note that, according to our experimental tests, moving from the 440M parameters to the 7B versions of BLOOM [Scao et al., 2022], FALCON [Almazrouei et al., 2023], or LLAMA-2 [Touvron et al., 2023b] averagely increases the decoding time per instance by a factor of x150. As a result, this would inadvertently multiply the required computational needs and runtimes. We thereby believe that incorporating LLMs and decoder-only architectures deserves an ad hoc extension. We know that the choice of the model may introduce bias in interpreting the results between different families. However, each AS type notoriously requires specific adaptions. Thus, we opt for the best representative models for each type, following BART-inspired architectures.

Additional Datasets Regarding testbeds, we carefully considered two well-known representative public datasets per AS type; further analysis should contemplate additional datasets from each AS family (e.g., BILLSUM [Kornilova and Eidelman, 2019] for SDS and GOVREPORT [Huang et al., 2021] for LDS). Future work should investigate other AS fields, such as dialogue summarization [Feng et al., 2022b], and other NLG tasks, such as question answering.

Multi-Lingual Resources The experimental analysis is focused solely on English. However, it is important to recognize that this choice was influenced by several factors, including the availability of datasets and fine-tuned models in the AS settings analyzed. English serves as a lingua franca in many domains, achieving a broader impact and enabling a substantial body of research to be conducted effectively due to the wealth of resources. Shifting towards multi-lingual insights necessitates an entirely new set of models and datasets for each AS type, thus doubling the number of runs and computational days. We plan to extend our analysis to multi-lingual settings in future work.

Increased Sample Size We recognize that we used a limited set of samples in comparative experiments. Testing all delineated configurations on all entire datasets would inadvertently multiply the required computational needs and runtimes, becoming unsustainable with our hardware configuration (Appendix 6.2.E). To mitigate this problem, we performed stratified sampling, reflecting the population rightly and avoiding common pure random sampling.

Novel Decoding Strategy This research centers on a thorough comparison of existing decoding methods alongside exhaustive hyperparameter permutations and evaluation axes. The goal is to grasp the impact of well-established decoding strategies on AS tasks from a neutral point of view and provide fresh insights to the community. In contrast to previous work, such as Top-p Sampling [Holtzman et al., 2020] and Contrastive Search [Su et al., 2022], we avoid presenting another decoding technique with a shallow evaluation related to affected quality attributes (e.g., naturalness and diversity for Top-p, coherence for Contrastive Search). For these reasons, the introduction of a new heuristic falls beyond the scope of our study in favor of a horizontal assessment on an unprecedented scale.

Diversity Metrics We acknowledge that there are multiple ways to summarize a document and that the lack of diversity in decoded solutions is crippling in the NLG sector, which highlights a relevant development for our paper. However, the quality and diversity criteria are not equally important in all tasks, where improving one often comes at the cost of the other [Aralikatte et al., 2021]. According to Meister et al. [2022], we argue that AS goals place a higher premium on accurate outputs than alternative ones. This factor is opposite to open-ended domains and tasks such as dialogue generation, aiming to maintain an enjoyable conversation with a human partner, avoiding repetitions.

Unified Metric Unlike other benchmarks such as GLUE [Wang et al., 2019b] and SuperGLUE [Wang et al., 2019a], this study uses metrics to evaluate orthogonal quality dimensions in generated summaries, from prediction-target semantic similarity to source-target compression and actuality. These aspects encompass different goals, boundaries, and interpretation keys, which make it not viable or sound to establish a single overall score. It is worth noting that, based on our analysis of the literature, no previous research has computed a set of 10 distinct NLG metrics to assess artificial summaries, as we have in Prism. We hope that our work will help researchers grapple with the identification of the finest decoding options for AS and set the stage for a new wave of heuristics.

TABLE 6.7. A summary of the decoding strategies benchmarked in this abstractive summarization study.

DECODING STRATEGIES[*,†]

Greedy Search

It selects the most probable token at each t (locally-optimal decision):

$$y_t = \underset{y \in \bar{\mathcal{V}}}{\operatorname{argmax}} \log p_\theta(y \mid \boldsymbol{x}, \boldsymbol{y}_{<t}) \quad \text{(for } t > 0\text{)}$$

Contrastive Search [Su et al., 2022]

It extends Greedy Search by jointly considering the model confidence (i.e., semantic coherence) and the similarity w.r.t. the previous context:

$$y_t = \underset{y \in \bar{\mathcal{V}}^{(k)}}{\operatorname{argmax}} \{(1-\alpha) \times p_\theta(y|\boldsymbol{x}, \boldsymbol{y}_{<t}) - \alpha \times (\max\{s(h_y, h_{y_j}) : 1 \le j \le t-1\})\} \quad \text{(for } t > 0\text{)}$$

$\bar{\mathcal{V}}^{(k)}$ is the set of top-k predictions from the LM's probability distribution ($k \in \mathbb{Z}_+$). The second term is a degeneration penalty governed by $\alpha \in [0, 1]$; it is computed with the cosine similarity $s(\cdot, \cdot)$ between the candidate representation h_y and the prior tokens. When $\alpha = 0$, Contrastive Search degenerates to Greedy Search.

Beam Search [Lowerre, 1976]

A pruned breadth-first search algorithm that expands the hypotheses in a greedy left-to-right way, retaining the top-b candidates at each t:

$$Y_t = \underset{\substack{Y' \subseteq \mathcal{B}_t, \\ |Y'|=b}}{\operatorname{argmax}} \mathcal{S}(Y'_t) \quad \text{(for } t > 0\text{)}$$

$\mathcal{B}_t$ denotes all possible roll-out extensions $\{\boldsymbol{y}_{t-1} \circ y \mid y \in \bar{\mathcal{V}}$ **and** $\boldsymbol{y}_{t-1} \in Y_{t-1}\}$. $\mathcal{S} : \mathcal{Y} \to \mathbb{R}$ is a scoring function on $Y \subseteq \mathcal{Y}$. By default, $\mathcal{S}(Y) = \sum_{\boldsymbol{y} \in Y} \log p_\theta(\boldsymbol{y}|\boldsymbol{x})$. $\boldsymbol{y}*$ is chosen from the final set Y_T.

Diverse Beam Search [Vijayakumar et al., 2018]

In vanilla Beam Search, as t and b increase, the candidate summaries of a single beam gradually occupy the top positions. This inner functioning results in a high overlap among hypotheses that often only differ by punctuation and slight morphological variations [Han et al., 2019], indicating poor search space coverage. Diverse Beam Search splits a beam into sub-groups sequentially optimized and adds a penalty in $\mathcal{S}$ to encourage inter-group dissimilarity:

$$\mathcal{S}(Y_t^{(g)}) = \sum_{\boldsymbol{y} \in Y} \log p_\theta(\boldsymbol{y}|\boldsymbol{x}) - \lambda \sum_{g' < g} \Delta\left(\boldsymbol{y}, Y_t^{(g')}\right)$$

$\Delta(\boldsymbol{y}, Y_t^{(g')})$ is a diversity measure whose strength is controlled by λ, and g is a sub-group.

Sampling

Instead of approximating $\boldsymbol{y}^*$, it makes a random selection at each t, thus giving a non-zero chance to every token following the model distribution ($\sim$):

$$y_t \sim p_\theta(\cdot|\boldsymbol{x}, \boldsymbol{y}_{<t}) \quad \text{(for } t > 0\text{)}$$

Softmax can be re-calibrated through a temperature parameter $\tau \in (0, 1]$. By decreasing τ (1 = no effect), we skew the distribution towards likely tokens and reduce the mass in the unreliable tail:

$$p_\theta(y|\boldsymbol{x}, \boldsymbol{y}_{<t}) = \frac{\exp(\frac{u}{\tau})}{\sum_j \exp(\frac{u_j}{\tau})}$$

where $u \in U$ are logits.

Top-k Sampling [Fan et al., 2018]

It limits the sampling space to the top-k probable tokens, regardless of the distribution shape. Here, the decoding maximizes:

$$\sum_{y \in \bar{\mathcal{V}}^{(k)}} p_\theta(y|\boldsymbol{x}, \boldsymbol{y}_{<t})$$

where $\bar{V}^{(k)} \subseteq \bar{V}$.

Top-p (Nucleus) Sampling [Holtzman et al., 2020]

Instead of assuming a fixed-sized shortlist, it dynamically expands and contracts the candidate pool according to the shape of the distribution (e.g., fewer contenders if sharp). When many next tokens are plausible, the allowed set reflects that. Formally, it samples from the smallest subset of tokens whose cumulative probability mass exceeds a chosen threshold $p \in (0, 1]$:

$$\sum_{y \in \bar{\mathcal{V}}^{(p)}} p_\theta(y|\boldsymbol{x}, \boldsymbol{y}_{<t}) \ge p$$

Beam Sampling [Caccia et al., 2020]

It is a variant of Beam Search where, at each t, the b next tokens for constructing $\mathcal{B}$ are sampled conditioned on the current hypotheses (no local optimum).

η Sampling [Hewitt et al., 2022]

It samples tokens above an entropy-dependent probability threshold to avoid unnecessary cutting-offs. Eligible tokens come from:

$$\{y \in \bar{\mathcal{V}} \mid p_\theta(y|\boldsymbol{x}, \boldsymbol{y}_{<t}) > \eta\}$$
$$\eta = \min\left(\epsilon, \alpha \exp(-e_{\boldsymbol{x}, \boldsymbol{y}_{<t}})\right)$$

where $\exp(-e_{\boldsymbol{x}, \boldsymbol{y}_{<t}})$ is the expected next token probability given the entropy $e_{\boldsymbol{x}, \boldsymbol{y}_{<t}}$, scaled by a constant α. To expose a single hyperparameter, α is set to $\sqrt{\epsilon}$. The general principle is to only truncate tokens whose probabilities are low relative to the rest of the distribution or an absolute threshold.

[*] Basic notation. $\boldsymbol{x}$ = document(s), $\boldsymbol{y}$ = summary, θ = model weights, t = decoding step, $\bar{\mathcal{V}} := \mathcal{V} \cup \{\text{EOS}\}$, p_θ = model probability distribution, $\mathcal{Y}$ = hypothetical summaries, b = beam size, $\boldsymbol{y}*$ = best summary.
[†] ● = deterministic, ● = stochastic.

TABLE 6.8. Hyperparameters for each decoding strategy.

HYPERPARAMS*,†	Deterministic				Stochastic				
	Greedy	Contrastive	Beam Search	Diverse Beam Search	Sampling	Top-k Sampling	Top-p Sampling	Beam Sampling	η Sampling
no_rep_ngram_size	{ 2, 3, 4, 5 }								
early_stopping			✓	✓				✓	
diversity_penalty				0.{2, 4, 6, 8}, 1.0					
num_beams			[2, 3, …, 10]	[2, 3, …, 10]				[2, 3, …, 10]	
temperature					0.{8, 9}, 1.0	0.{8, 9}, 1.0	0.{8, 9}, 1.0		
top_k		[20, 30, …, 60]				[20, 30, …, 60]	{/, 20, …, 60}		
top_p							0.{4, 6, 8}	0.9	
penalty_alpha		0.{2, 4, 6, 8}, 1.0							
eta_cutoff									$\{4,2\}\times10^{-3}$, $\{9,6,3\}\times10^{-4}$
do_sample					✓	✓	✓	✓	✓
# Runs	16	400	144	720	48	240	864	144	80

* {…} = sets, […] = series, ✓ = true, / = none.
† orange and cyan initialization schemes are tested on 2 and 4 datasets, respectively; all other values are equally applied to all the 6 surveyed datasets.

TABLE 6.9. A summary of the metrics used in this study to assess the generated summaries.

METRICS*,†

ROUGE [Lin, 2004] $[0,1],\uparrow$

Unigrams (r_1), bigrams (r_2), and longest common subsequence ($r_{L\text{sum}}$) lexical overlaps (%) between the inferred and gold summaries, i.e., a proxy for informativeness and fluency. Inspired by Moro et al. [2023b], we measure an aggregated judgment:

$$\mathcal{R} = \frac{\text{avg}(r_1, r_2, r_{L\text{sum}})}{1+\sigma_r^2} \quad (6.8)$$

where σ_r^2 is the variance of 0-1 normalized ROUGE F1 scores. $\mathcal{R}$ penalizes model results with discrepant r_1, r_2, $r_{L\text{sum}}$ values.

BERTScore [Zhang et al., 2020b] $[-1,1],\uparrow$

Semantic recall formalized as:

$$\frac{\sum_{y_i \in \boldsymbol{y}} \text{idf}(y_i) \max_{\hat{y}_j \in \hat{\boldsymbol{y}}} \mathbf{e}(y_i)^\top \mathbf{e}(y_j)}{\sum_{y_i \in \boldsymbol{y}} \text{idf}(y_i)}$$

where e(·) is a BERT [Devlin et al., 2019] token embedding.

Perplexity [Jelinek et al., 1977] $[0,\infty[,\downarrow$

The naturalness of a summary w.r.t. the data seen by a model (GPT-2 [Radford et al., 2019b] in this paper) during training:

$$ppl(\boldsymbol{y}) = \exp\left(\frac{1}{t}\sum_{i}^{t} \log p_\theta(y_i|\boldsymbol{y}_{<i})\right)$$

.

Coverage [Grusky et al., 2018] $[0,1],\uparrow$

The percentage of summary words within the source text:

$$\frac{1}{|\boldsymbol{y}|}\sum_{\boldsymbol{f}\in\mathcal{F}(\boldsymbol{x},\boldsymbol{y})} |\boldsymbol{f}|$$

where $\mathcal{F}$ is the set of all fragments, i.e., extractive character sequences. When low, it suggests a high chance for unsupported entities and facts.

Density [Grusky et al., 2018] $[0,|x|_c],\downarrow$

The average length of the extractive fragments. It is formulated as:

$$\frac{1}{|\boldsymbol{y}|_c}\sum_{\boldsymbol{f}\in\mathcal{F}(\boldsymbol{x},\boldsymbol{y})} |\boldsymbol{f}|_c^2$$

where $||_c$ is the character length. When low, it suggests that most summary sentences are not verbatim extractions from the sources (abstractive).

Compression [Grusky et al., 2018] $[0,|x|],\uparrow$

The document-summary word ratio: $|\boldsymbol{x}|/|\boldsymbol{y}|$.

Unique N-gram Ratio (UNR) [Xiao and Carenini, 2020] $[0,1],\uparrow$

The summary n-grams uniqueness:

$$\frac{\text{count}(\text{uniq_n_gram}(\boldsymbol{y}))}{\text{count}(\text{n_gram}(\boldsymbol{y}))}$$

where we take n from $[1,3]$ and divide the average by variance.

Normalized Inverse of Diversity (NID) [Xiao and Carenini, 2020] $[0,1],\downarrow$

It reckons redundancy by inverting the entropy of summary unigrams and applying length normalization:

$$1 - \text{entropy}(\boldsymbol{y})/\log(|\boldsymbol{y}|)$$

BARTScore-F [Yuan et al., 2021] $[-\infty,0],\uparrow$

The weights θ of a pretrained BART model [Lewis et al., 2020a] are used to estimate how likely hypothesis and reference are reciprocal paraphrases (i.e., probability of generating one giving the other):

$$\sum_{t=1}^{|\boldsymbol{y}|} \log p(y_t|\boldsymbol{y}_{<t}, \boldsymbol{x}, \theta_B)$$

We measure faithfulness by mapping sources and predicted summaries to $\boldsymbol{x}$ and $\boldsymbol{y}$.

Carburacy [Moro et al., 2023b] $[0,1],\uparrow$

Carbon-aware accuracy measure modeling both the AS model effectiveness and eco-sustainability: $\Upsilon = \left(\exp^{\log_\alpha \mathcal{R}}\right)/1+\mathcal{C}\cdot\beta$ where $\mathcal{R}$ is defined in Equation 6.8, $\mathcal{C}$ is the kg of CO_2 emissions produced by the model to process a single instance x at inference time, α and β are trade-off hyperparameters.

* Blue text indicates the bound value and the general reading key (i.e., ↑ = higher is better, ↓ = lower is better).
† ● = reference-based (gold-summary dependence), ● = reference-free

TABLE 6.10. Dataset statistics before and after sampling, including test set size, number of sources per instance, and the words in the source and target texts. All values are averaged except "# Samples."

	Before sampling		Source		Target
	Dataset	**# Samples**	# Docs	# Words	# Words
SDS	XSUM	11,333	1	421.2	22.9
	CNN/DM	11,490	1	761.3	57.4
LDS	PUBMED	6658	1	3070.4	206.7
	ARXIV	6440	1	5733.2	161.6
MDS	MULTI-NEWS	5621	2.8	2026.3	245.9
	MULTI-LEXSUM	616	10.3	88,122.9	126.5

	After sampling		Source		Target
	Dataset	**# Samples**	# Docs	# Words	# Words
SDS	XSUM	1134	1	414.1	22.8
	CNN/DM	1148	1	753.8	58.0
LDS	PUBMED	667	1	3095.5	207.2
	ARXIV	644	1	5770.7	160.6
MDS	MULTI-NEWS	555	2.8	1947.9	248.0
	MULTI-LEXSUM	616	10.3	88,122.9	126.5

TABLE 6.11. List of the utilized datasets and relative length constraints.

Dataset	**URL**	**Input**	**Output**	
		Max Len	Min Len	Max Len
XSUM	https://huggingface.co/datasets/xsum	1024	20	100
CNN/DM	https://huggingface.co/datasets/cnn_dailymail	1024	20	100
PUBMED	https://huggingface.co/datasets/ccdv/pubmed-summarization	4096	100	256
ARXIV	https://huggingface.co/datasets/ccdv/arxiv-summarization	4096	100	256
MULTI-NEWS	https://huggingface.co/datasets/multi_news	4096	100	256
MULTI-LEXSUM	https://huggingface.co/datasets/allenai/multi_lexsum	4096	50	256

TABLE 6.12. Related work on decoding strategies comparison.

Source	Decoding Strategies	Hyperparameters*	Task**	Automatic Metrics	Systematic Benchmark
Holtzman et al. [2020]	Greedy Search	/	OEG	Perplexity, Self-BLEU, Repetition, Zipf Coefficient	✗
	Beam Search	$b \in \{4, 8, 16\}$			
	Sampling	$\tau \in [0.1, 0.2, \ldots, 1.0]$			
	Top-k Sampling	$k \in 5 \times 2^{[1,2,\ldots,12]}$			
	Top-p Sampling	$\approx p \in [0.1, 0.2, \ldots, 0.9, 0.95]$			
Meister et al. [2022]	Greedy Search	/	OEG, MT, SDS, DG, SG	BLEU, COMET, ROUGE BLEURT, DIST-n, ENT-n, n-GRAM DIV., Self-BLEU, Repetition	✓
	Beam Search	$b \in \{5, 10\}$			
	Diverse Beam Search	Δ = Hamming, $\lambda = 0.7$, $\lvert g\rvert = b = 5$			
	Sampling	/			
	Top-k Sampling	$k = 30$			
	Top-p Sampling	$p = 0.85$			
Ippolito et al. [2019]	Beam Search	$b = 10$	DG, IC	Perplexity, DIST-n, ENT-n, SPICE	✓
	Diverse Beam Search	Δ = Hamming, $\lambda = 0.8$, $\lvert g\rvert = b = 10$			
	NPAD Beam Search	$b = 10$, $\sigma_0 = 0.3$			
	Clustered Beam Search	$b = 10$, $c = 5$			
	Top-g Capping Beam Search	$b = 10$, $g_c = 3$			
	Iterative Beam Search	$b = 10$, $i = 5$			
	Sampling	$\tau \in \{0.5, 0.7, 1.0\}$			
Leblond et al. [2021]	Greedy	/	MT	BLEU, BERTScore	✗
	Beam Search	$b \in [2, 4, \ldots, 10, 20]$, logits $\tau \in [0.6, 0.8, \ldots, 1.4]$ normalization $\tau \in [0.4, 0.6, \ldots, 1.0]$			
	Value-Guided Beam Search	$b = 6$, logits $\tau \in [0.6, 0.8, \ldots, 1.4]$, $\alpha_{lc} \in [0, 0.1, \ldots, 0.9, 0.95, 1.0]$			
	Monte Carlo Tree Search	logits $\tau \in \{0.9, 1.1, 1.3\}$, $c_{puct} \in [1.0, 2.0, \ldots, 6.0, 8.0]$			
	Sampling + Ranking Variants	$\tau \in [0.15, 0.25, \ldots, 0.95]$			
Su et al. [2022] Su and Collier [2023]	Greedy	/	OEG, DG, SDS, CG, MT	Perplexity, Accuracy, Repetition, n-GRAM DIV., MAUVE, Semantic Coherence	✗
	Contrastive Search	$k \in [2, 3, \ldots, 10]$, $\alpha \in [0.1, 0.2, \ldots, 1]$			
	Beam Search	$b \in \{4, 5, 10\}$			
	Typical Sampling	$\tau = 0.95$			
	Top-k Sampling	$k = 50$			
	Top-p Sampling	$p = 0.95$			
Ours	Greedy	no_repetition_ngram_size $\in \{2, 3, 4, 5\}$	SDS, LDS, MDS	ROUGE, BERTScore, Perplexity, Coverage, Density, Compression, UNR, NID, BARTScore-F, Carburacy + Runtime	✓
	Contrastive Search	no_repetition_ngram_size $\in \{2, 3, 4, 5\}$, $k \in [20, 30, \ldots, 60]$, $\alpha \in [0.2, 0.4, \ldots, 1]$			
	Beam Search	no_repetition_ngram_size $\in \{2, 3, 4, 5\}$, $b \in [2, 3, \ldots, 10]$			
	Diverse Beam Search	no_repetition_ngram_size $\in \{2, 3, 4, 5\}$, Δ = Hamming, $\lambda \in [0.2, 0.4, \ldots, 1.0]$, $\lvert g\rvert = b \in [2, 3, \ldots, 10]$			
	Sampling	no_repetition_ngram_size $\in \{2, 3, 4, 5\}$, $\tau \in \{0.8, 0.9, 1.0\}$			
	Top-k Sampling	no_repetition_ngram_size $\in \{2, 3, 4, 5\}$, $\tau \in \{0.8, 0.9, 1.0\}$, $k \in [20, 30, \ldots, 60]$			
	Top-p Sampling	no_repetition_ngram_size $\in \{2, 3, 4, 5\}$, $\tau \in \{0.8, 0.9, 1.0\}$, $k \in [\text{None}, 20, 30, \ldots, 60]$ $p \in 0.4, 0.6, 0.8$			
	Beam Sampling	no_repetition_ngram_size $\in \{2, 3, 4, 5\}$, $b \in [2, 3, \ldots, 10]$, $p = 0.9$			
	η Sampling	no_repetition_ngram_size $\in \{2, 3, 4, 5\}$, $\eta \in \{0.0003, 0.0006, 0.0009, 0.002, 0.004\}$			

* b = beam size, τ = temperature, k/p = top-k/p thresholds for selecting candidate pools, Δ = diversity measure, λ = diversity penalty, $\lvert g\rvert$ = number of sub-groups, σ_0 = decoder-level random noise, c = number of clusters, g_c = top parent hypotheses considered at each time step, i = number of iterations, α_{lc} = linear combination weights, η = entropy-dependent cutting-off.
** OEG = Open-Ended (Unconditional) Generation, MT = Machine Translation, DG = Dialogue Generation, SG = Story Generation, IC = Image Captioning, CG = Code Generation, SDS = Short Document Summarization, LDS = Long Document Summarization, MDS = Multi-Document Summarization.

TABLE 6.13. List of the utilized models.

Model	Dataset	URL
BART-large	XSUM	https://huggingface.co/facebook/bart-large-xsum
	CNN/DM	https://huggingface.co/facebook/bart-large-cnn
LED-large	PUBMED	https://huggingface.co/patrickvonplaten/led-large-16384-pubmed
	ARXIV	https://huggingface.co/allenai/led-large-16384-arxiv
PRIMERA-large	MULTI-NEWS	https://huggingface.co/allenai/PRIMERA-multinews
	MULTI-LEXSUM	https://huggingface.co/allenai/primera-multi_lexsum-source-short

TABLE 6.15. Average FLOPs computational performance of the evaluated decoding strategies.

Greedy	**Contrastive**	**Beam Search**	**DBS**	**Sampling**	**Top-k**	**Top-p**	**η Sampling**	**Beam Sampling**
1.7e17	1.7e17	1.9e17	2.6e17	2.1e17	1.8e17	2.4e17	2.6e17	1.8e17

TABLE 6.16. Example attached to annotation instructions.

Document
African elephants are the largest land animals on Earth. They are known for their distinctive large ears and tusks, which are actually elongated incisor teeth. These gentle giants live in herds led by a matriarch and have a complex social structure. African elephants face threats from poaching and habitat loss, which have led to a decline in their population.
Gold Summary
African elephants live in herds led by a matriarch. Despite their gentle nature, they face endangerment due to poaching and habitat loss, leading to a declining population. **Topics**: African elephants - Social structure - Threats they face.
Summary 1 (High Recall, High Precision, Low Faithfulness)
African elephants live in herds led by a matriarch. They are not endangered and face no threats. **Explanation**: Summary 1 has a *high Recall* because it includes all relevant information/topics of the gold summary; it has a *high Precision* because it includes only the relevant information/topics without adding irrelavant details; the *Faithfulness is low* because it includes incorrect information about their endangerment status.
Summary 2 (High Recall, Low Precision, Low Faithfulness)
African elephants, the Earth's largest land animals, are known for their large ears and tusks. They live individually and face threats from poaching and habitat loss, causing a decline in their population. **Explanation**: Summary 2 has a *high Recall* because it includes all relevant information/topics of the gold summary; it has a *low Precision* because it also includes additional information/topics that are not within the gold summary; the *Faithfulness is low* because it includes incorrect information about their social structure.

TABLE 6.17. Normalized tabulated scores categorized by dataset, metric, and decoding strategy. For each dataset, the decoding strategies are sorted by decreasing AVG.

Strategy	$\mathcal{R}$	BERTScore	Perplexity ↓	Coverage	Density ↓	Compression	UNR	NID ↓	BARTScore	Carburacy
XSum										
Beam Sampling	3.44	7.31	1.49	5.95	0.31	0.03	6.67	4.65	7.51	4.25
Greedy Search	3.25	8.05	1.28	6.28	0.35	0.04	5.96	6.22	8.11	4.24
Contrastive Search	3.27	8.01	1.37	6.17	0.32	0.04	5.97	6.16	8.09	4.27
η Sampling	3.27	8.01	1.40	6.08	0.35	0.03	5.96	6.17	8.10	4.26
Beam Search	3.30	7.97	1.30	6.27	0.35	0.04	5.85	6.37	8.06	4.28
Diverse Beam Search	3.80	6.82	1.65	5.96	0.28	0.05	4.48	8.21	7.25	4.80
Sampling	2.34	5.16	1.37	3.78	0.14	0.03	7.51	3.54	5.21	3.04
Top-k Sampling	2.22	5.48	1.37	4.23	0.23	0.03	7.09	4.25	5.45	2.90
Top-p Sampling	2.30	5.34	1.48	4.37	0.25	0.03	6.88	4.37	5.41	3.00
Cnn/Dm										
Greedy Search	8.28	9.88	6.55	9.94	5.22	0.07	7.33	4.40	9.91	8.77
η Sampling	8.25	9.86	6.24	9.70	5.26	0.07	7.32	4.40	9.89	8.76
Beam Search	8.30	9.87	6.49	9.98	5.58	0.08	7.42	4.40	9.93	8.80
Diverse Beam Search	8.44	9.90	6.12	9.86	6.12	0.08	7.30	4.49	9.80	8.92
Beam Sampling	8.20	9.79	6.36	9.71	5.28	0.08	7.36	4.38	9.84	8.48
Contrastive Search	8.26	9.87	6.27	9.93	6.20	0.09	7.31	4.40	9.89	8.77
Top-k Sampling	6.51	7.17	6.24	6.49	3.01	0.02	8.23	2.95	6.82	7.14
Sampling	6.46	7.12	6.30	6.49	3.08	0.02	8.20	2.94	6.79	6.99
Top-p Sampling	5.67	5.53	6.15	6.78	3.07	0.08	8.45	2.27	5.02	6.37
PubMed										
Diverse Beam Search	8.75	8.81	3.95	9.73	7.01	0.23	3.43	5.71	8.44	9.10
Contrastive Search	9.14	9.07	4.22	9.87	8.55	0.20	3.70	5.40	8.59	9.41
Greedy Search	9.11	9.06	4.25	9.86	8.89	0.19	3.82	5.34	8.65	9.31
Beam Search	8.97	8.99	4.17	9.86	9.13	0.19	3.98	5.28	8.71	9.26
Beam Sampling	9.04	9.03	4.23	9.73	8.88	0.19	3.85	5.34	8.62	9.24
η Sampling	9.13	9.07	4.23	9.75	8.90	0.18	3.71	5.39	8.59	9.27
Top-k Sampling	5.71	5.33	4.24	6.35	4.75	0.22	6.66	2.75	4.58	6.32
Sampling	5.74	5.33	4.32	6.34	4.71	0.21	6.64	2.75	4.57	6.33
Top-p Sampling	5.81	5.32	3.99	6.33	4.69	0.24	6.31	2.99	4.55	6.36
arXiv										
Diverse Beam Search	9.16	8.37	8.58	9.51	3.66	0.81	3.84	6.34	7.54	9.35
η Sampling	9.53	8.64	9.12	9.73	5.65	0.66	3.91	6.06	7.71	9.59
Contrastive Search	9.50	8.64	9.30	9.72	5.83	0.67	3.92	6.06	7.72	9.77
Beam Sampling	9.47	8.62	9.11	9.73	5.68	0.64	3.94	6.04	7.74	9.30
Greedy Search	9.45	8.59	9.32	9.73	6.62	0.65	3.86	6.02	7.85	9.62
Beam Search	9.37	8.57	9.29	9.79	7.14	0.63	3.90	5.97	7.91	9.55
Top-p Sampling	7.10	5.97	9.29	6.62	3.39	0.64	5.84	3.95	4.68	7.62
Sampling	7.02	6.05	9.40	6.64	3.82	0.62	6.11	3.78	4.76	7.56
Top-k Sampling	7.04	6.06	9.48	6.66	3.88	0.62	6.11	3.79	4.78	7.55
Multi-News										
Greedy Search	8.69	8.93	0.33	8.38	1.32	0.15	4.39	5.81	8.36	9.05
Beam Search	8.64	8.90	0.35	8.35	1.34	0.15	4.37	5.82	8.35	9.00
Contrastive Search	8.58	8.87	0.37	8.27	1.25	0.15	4.36	5.84	8.33	8.98
η Sampling	8.60	8.88	0.36	8.18	1.25	0.15	4.38	5.82	8.32	8.95
Beam Sampling	8.49	8.79	0.38	8.14	1.24	0.14	4.31	5.85	8.27	8.90
Diverse Beam Search	8.16	8.65	0.43	8.09	0.88	0.16	4.07	6.15	8.18	8.61
Top-k Sampling	6.13	5.69	0.37	5.36	0.75	0.14	6.54	3.50	4.90	6.77
Sampling	6.07	5.68	0.36	5.36	0.71	0.14	6.53	3.51	4.90	6.74
Top-p Sampling	6.14	5.55	0.36	5.40	0.74	0.14	6.36	3.60	4.81	6.80
Multi-LexSum										
Beam Search	7.66	7.99	0.18	9.40	0.42	8.75	3.33	5.93	8.10	8.25
Diverse Beam Search	7.32	7.96	0.12	7.52	0.43	9.19	3.58	6.15	7.85	7.98
Beam Sampling	7.59	7.97	0.08	7.70	0.40	8.45	3.20	5.93	8.07	8.05
Greedy Search	7.09	7.59	0.10	8.61	0.40	8.70	1.82	6.69	7.83	7.81
Contrastive Search	5.89	6.87	0.52	8.59	0.44	8.81	3.42	5.75	7.26	6.79
η Sampling	5.94	6.92	0.47	7.65	0.41	8.47	3.36	5.78	7.29	6.81
Top-p Sampling	6.86	7.47	0.26	5.92	0.23	8.46	2.15	6.51	7.73	7.61
Sampling	6.13	6.99	0.37	5.78	0.23	8.66	3.10	5.93	7.38	6.98
Top-k Sampling	6.12	7.02	0.42	5.75	0.22	8.33	3.16	5.93	7.38	6.98

TABLE 6.18. Best $\mathcal{R}$ hyperparameters for decoding strategies across all datasets.

Greedy Search	**Contrastive Search**	**Beam Search**
`no_repeat_ngram_size`=3	`no_repeat_ngram_size`=5, `top_k`=20, `penalty_alpha=0.2`	`no_repeat_ngram_size`=3, `num_beams`=9
Diverse Beam Search	**Sampling**	**Top-k Sampling**
`no_repeat_ngram_size`=3, `num_beams`=10, `diversity_penalty`=0.2	`no_repeat_ngram_size`=5, `temperature`=1.0	`no_repeat_ngram_size`=5, `temperature`=1.0 `top_k`=50
Top-p Sampling	**η Sampling**	**Beam Sampling**
`no_repeat_ngram_size`=3, `temperature`=1.0, `top_k`=20, `top_p`=0.4	`no_repeat_ngram_size`=5, `cut_off`=0.002	`no_repeat_ngram_size`=3, `num_beams`=10

TABLE 6.19. Hyperparameters for the top-3 $\mathcal{R}$ decoding strategies per dataset.

Strategy	`no_rep`	`num_beams`	`div_penalty`	`temperature`	`penalty_alpha`	`top_k`	`top_p`	`cutoff`
XSum								
1. Contrastive Search	2	-	-	-	1.0	20	-	-
2. Top-k Sampling	2	-	-	0.9	-	50	-	-
3. Top-p Sampling	2	-	-	0.9	-	50	0.6	-
Cnn/Dm								
1. Diverse Beam Search	3	5	0.2	-	-	-	-	-
2. Top-k Sampling	3	-	-	0.9	-	50	-	-
3. Top-p Sampling	3	-	-	0.9	-	-	0.4	-
PubMed								
1. Contrastive Search	5	-	-	-	0.8	30	-	-
2. Top-p Sampling	5	-	-	1.0	-	50	0.8	-
3. Beam Sampling	5	3	-	-	-	-	-	-
arXiv								
1. Diverse Beam Search	5	10	0.6	-	-	-	-	-
2. Top-p Sampling	5	-	-	1.0	-	20	0.6	-
3. η Sampling	5	-	-	-	-	-	-	0.0003
Multi-News								
1. Contrastive Search	5	-	-	-	0.2	30	-	-
2. Beam Search	4	10	-	-	-	-	-	-
3. Beam Sampling	5	9	-	-	-	-	-	-
Multi-LexSum								
1. Beam Search	3	8	-	-	-	-	-	-
2. Diverse Beam Search	3	7	0.2	-	-	-	-	-
3. Beam Sampling	3	8	-	-	-	-	-	-

TABLE 6.20. Hyperparameters effect grouped by datasets – Greedy Search.

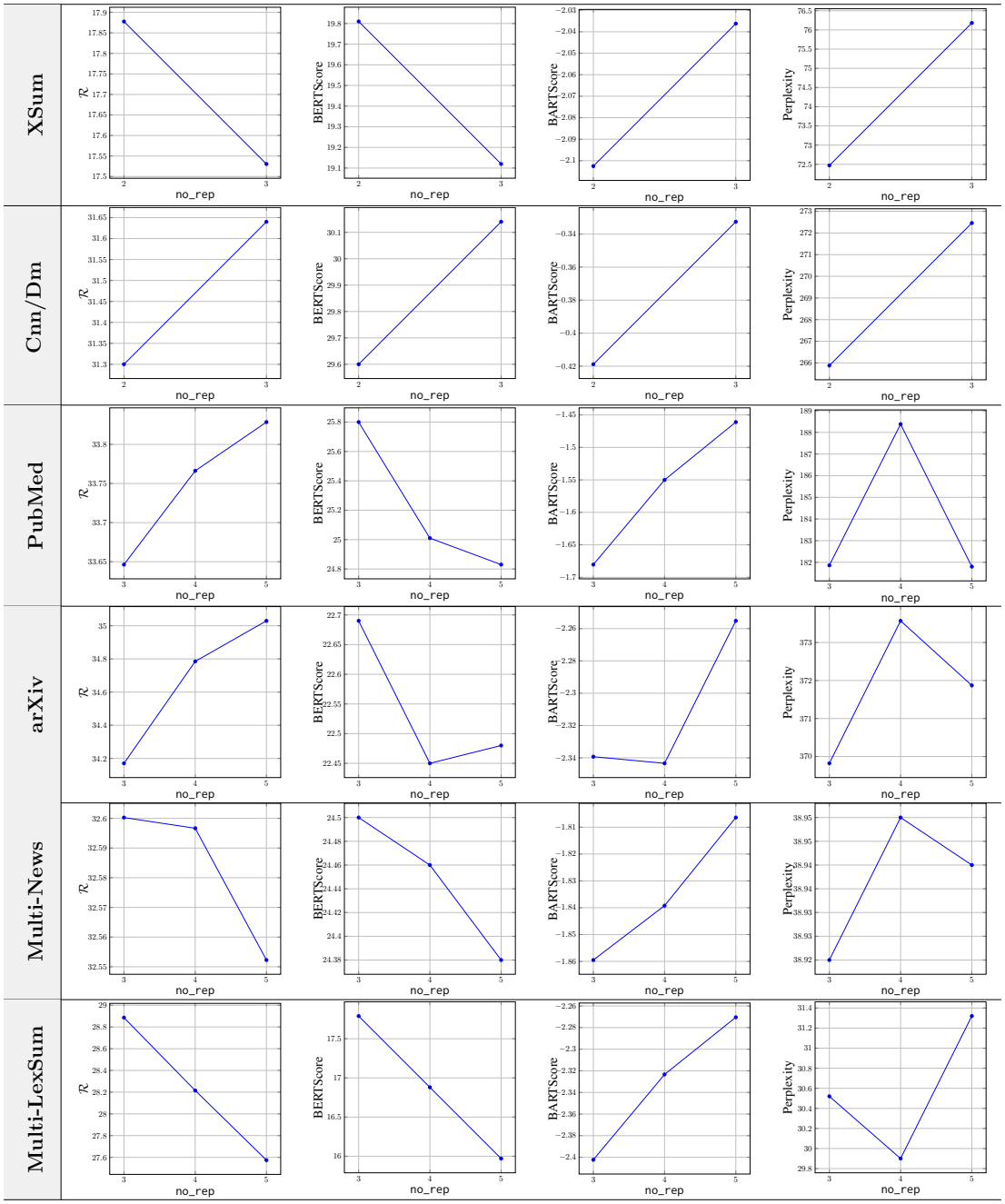


TABLE 6.21. Hyperparameters effect grouped by datasets – Contrastive Search.

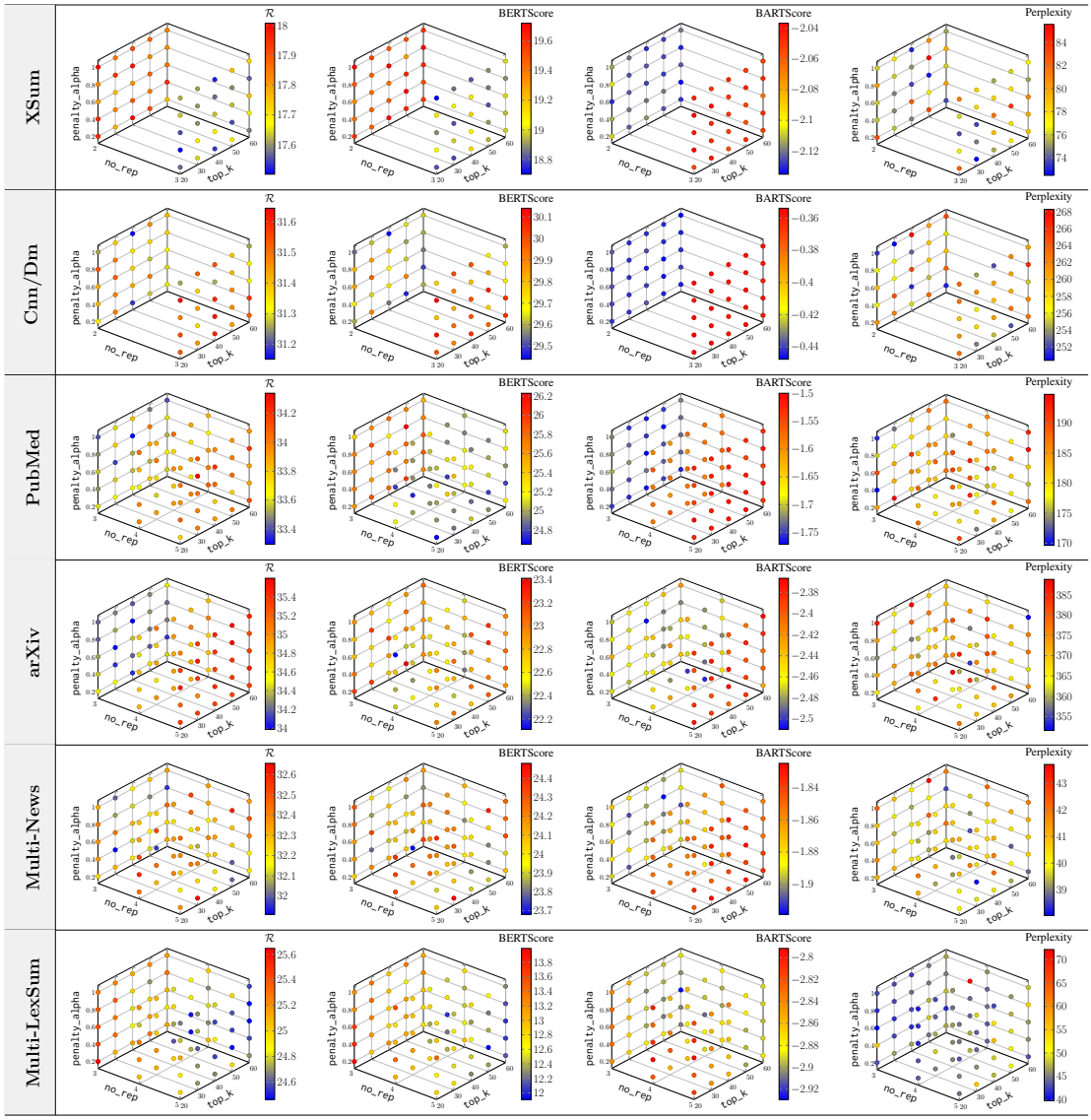


TABLE 6.22. Hyperparameters effect grouped by datasets – Beam Search.

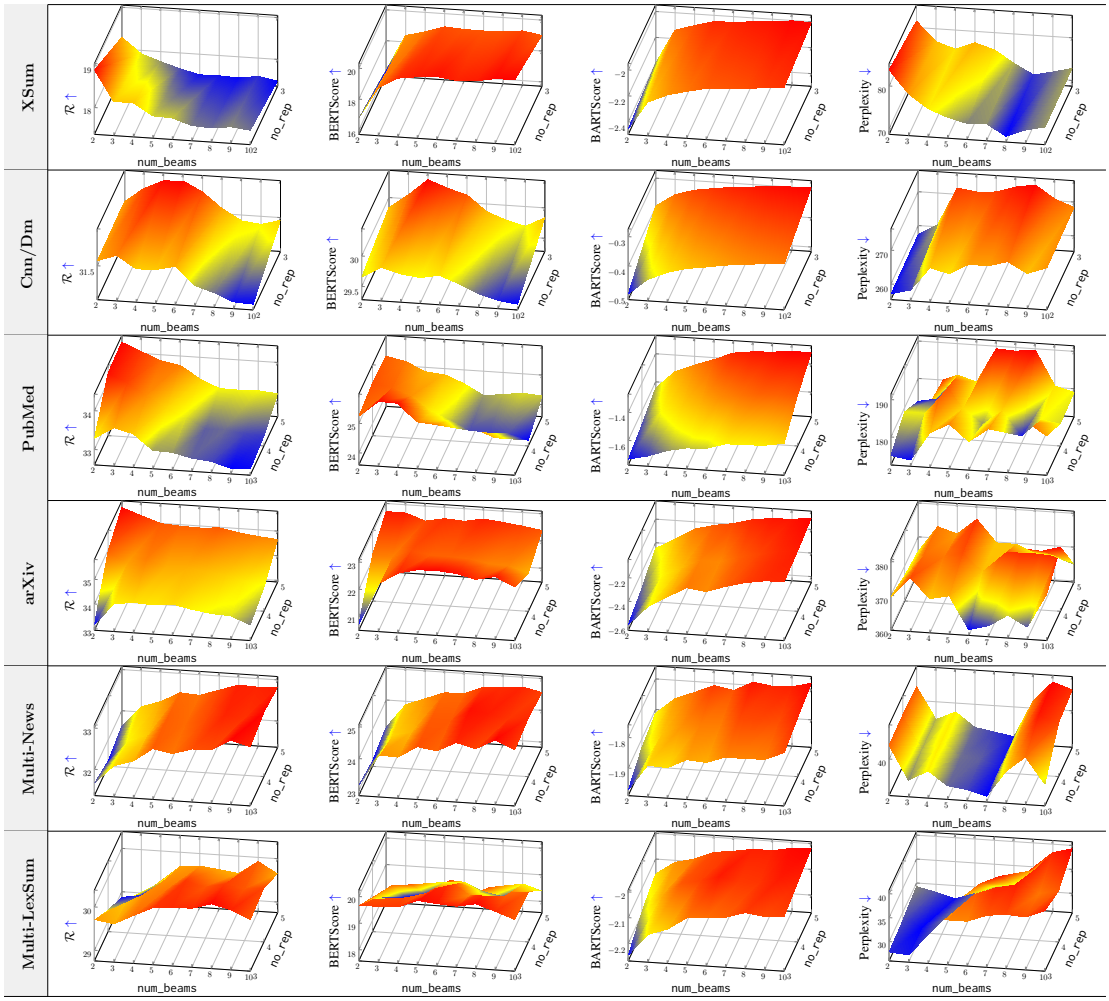


TABLE 6.23. Hyperparameters effect grouped by datasets – Diverse Beam Search.

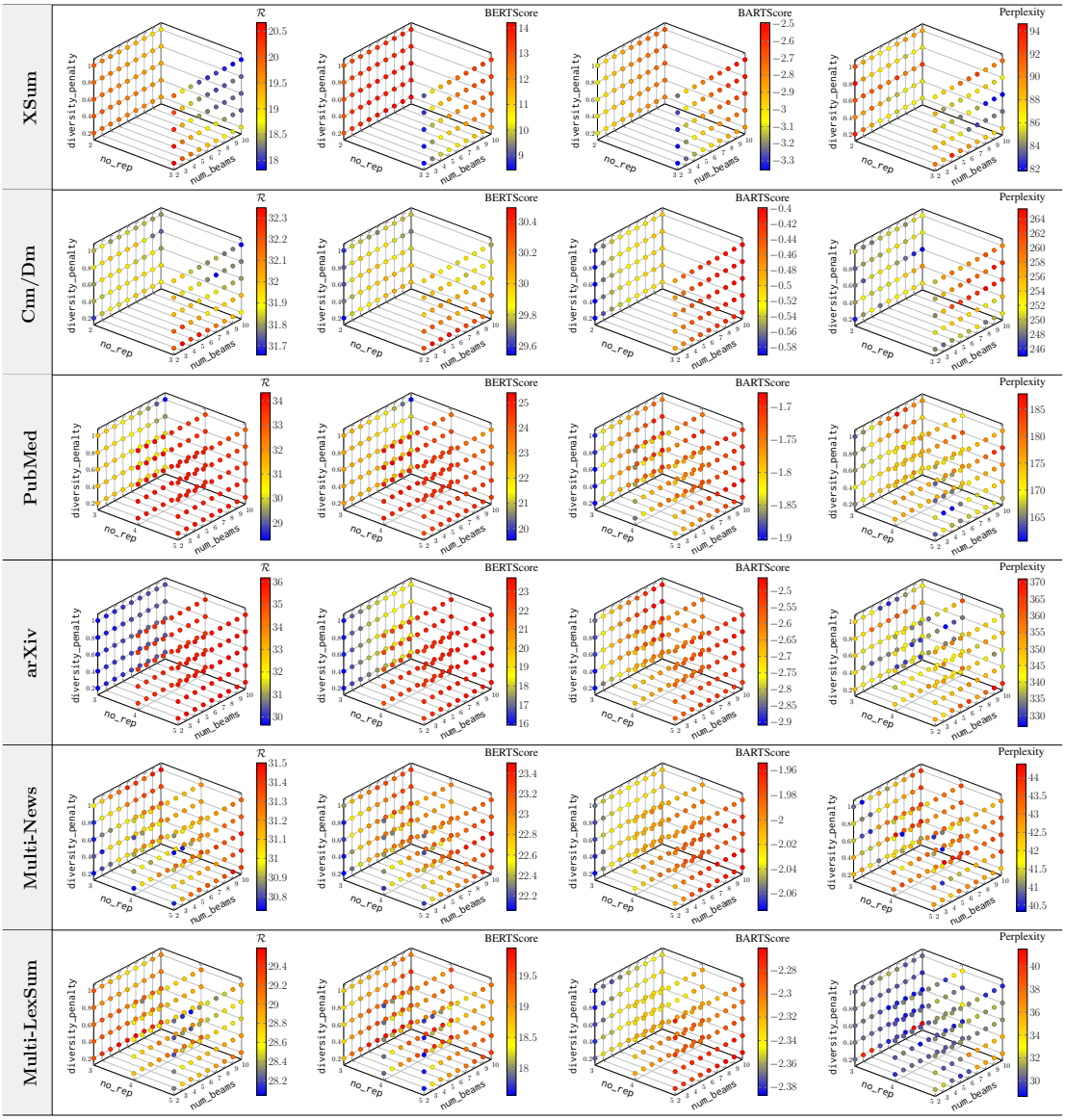


TABLE 6.24. Hyperparameters effect grouped by datasets – Sampling.

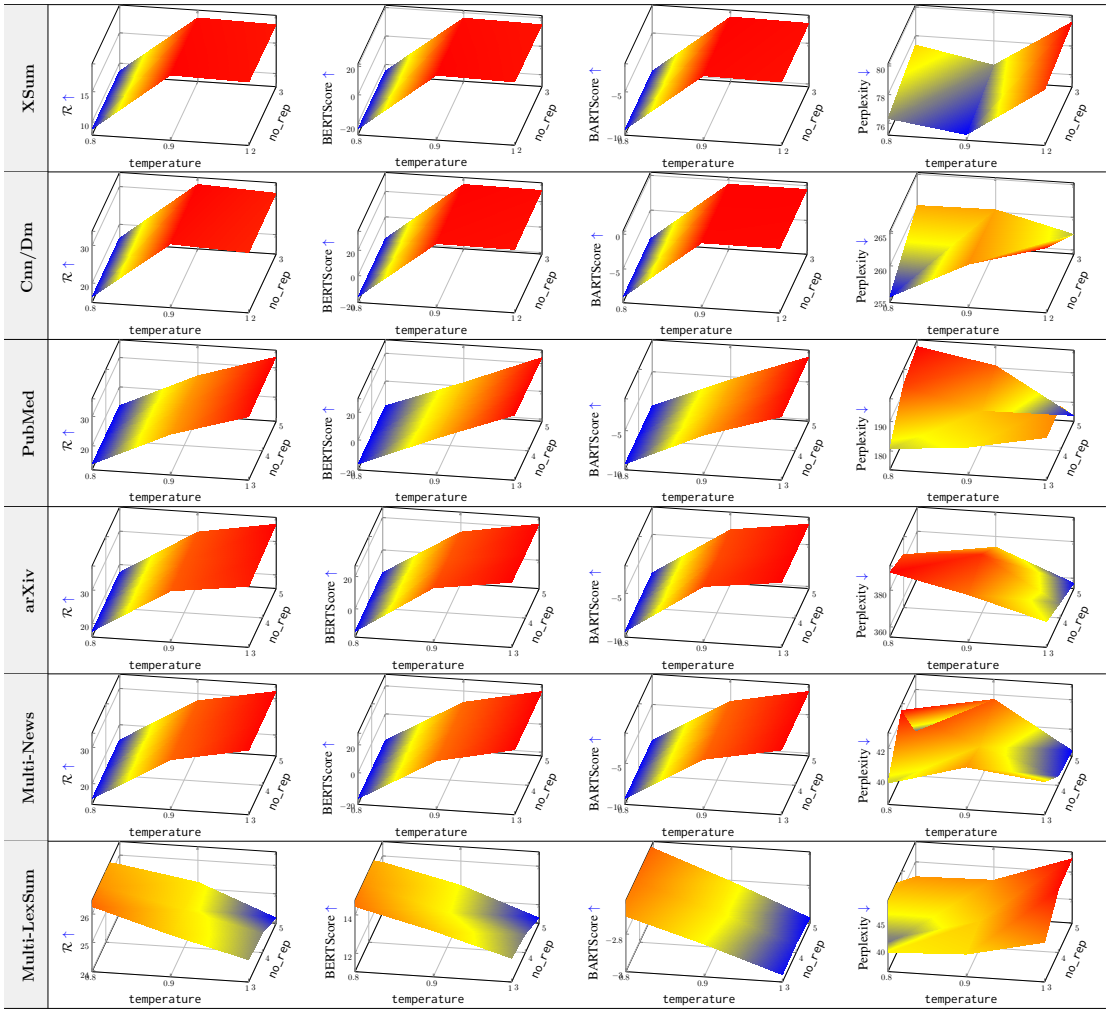


TABLE 6.25. Hyperparameters effect grouped by datasets – Top-k Sampling.

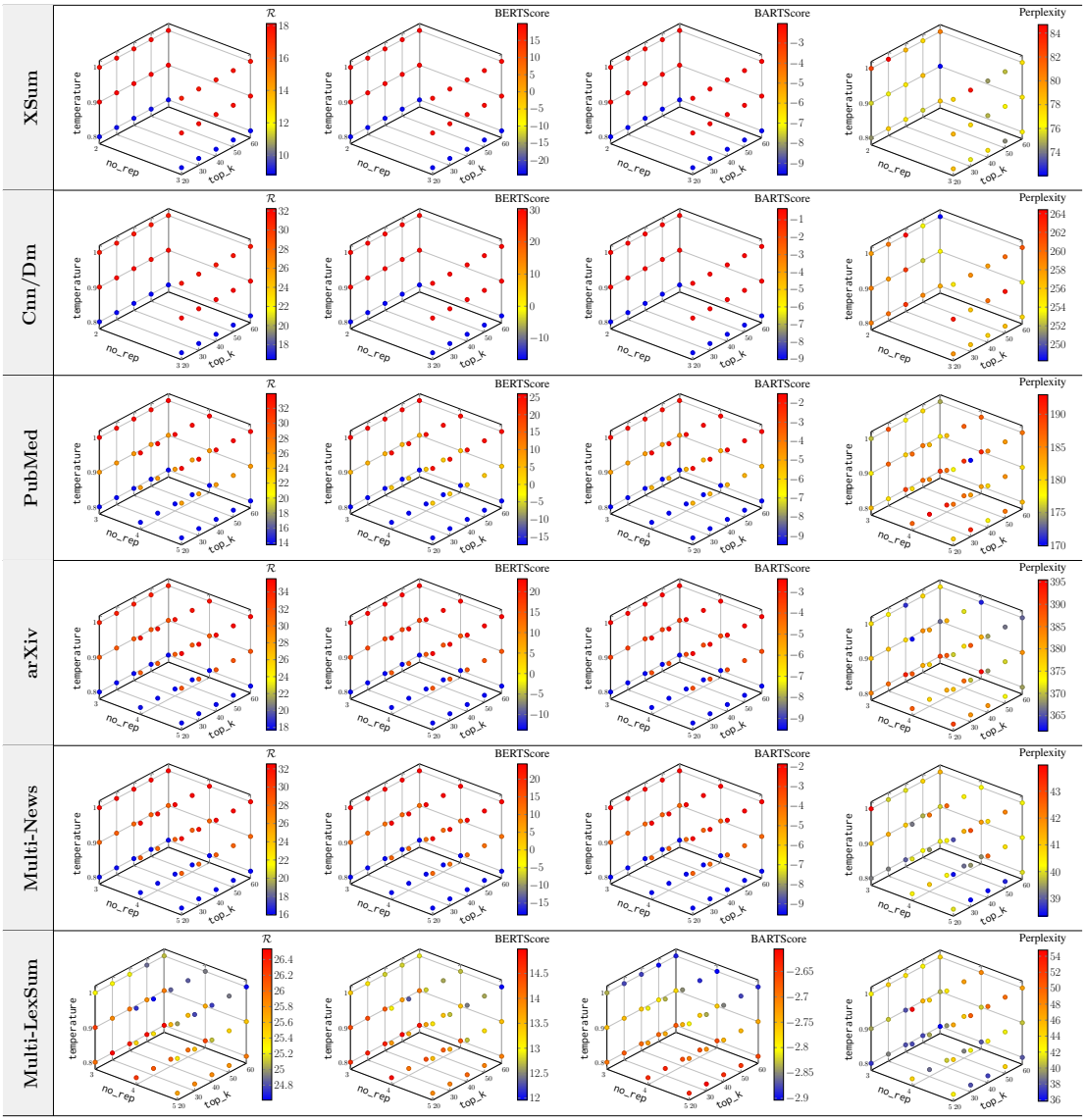


TABLE 6.26. Hyperparameters effect grouped by datasets – Top-p Sampling.

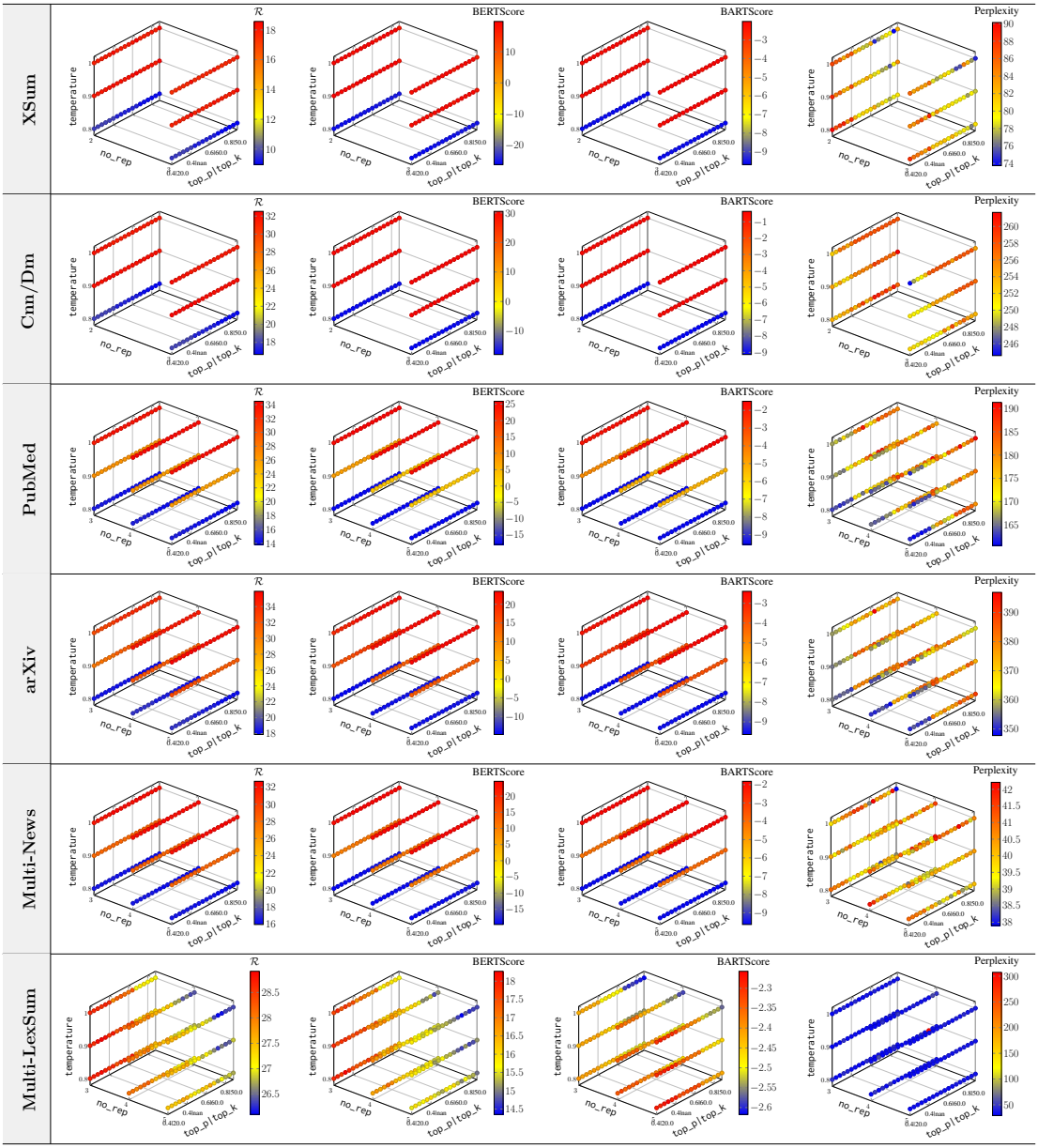


TABLE 6.27. Hyperparameters effect grouped by datasets – η Sampling.

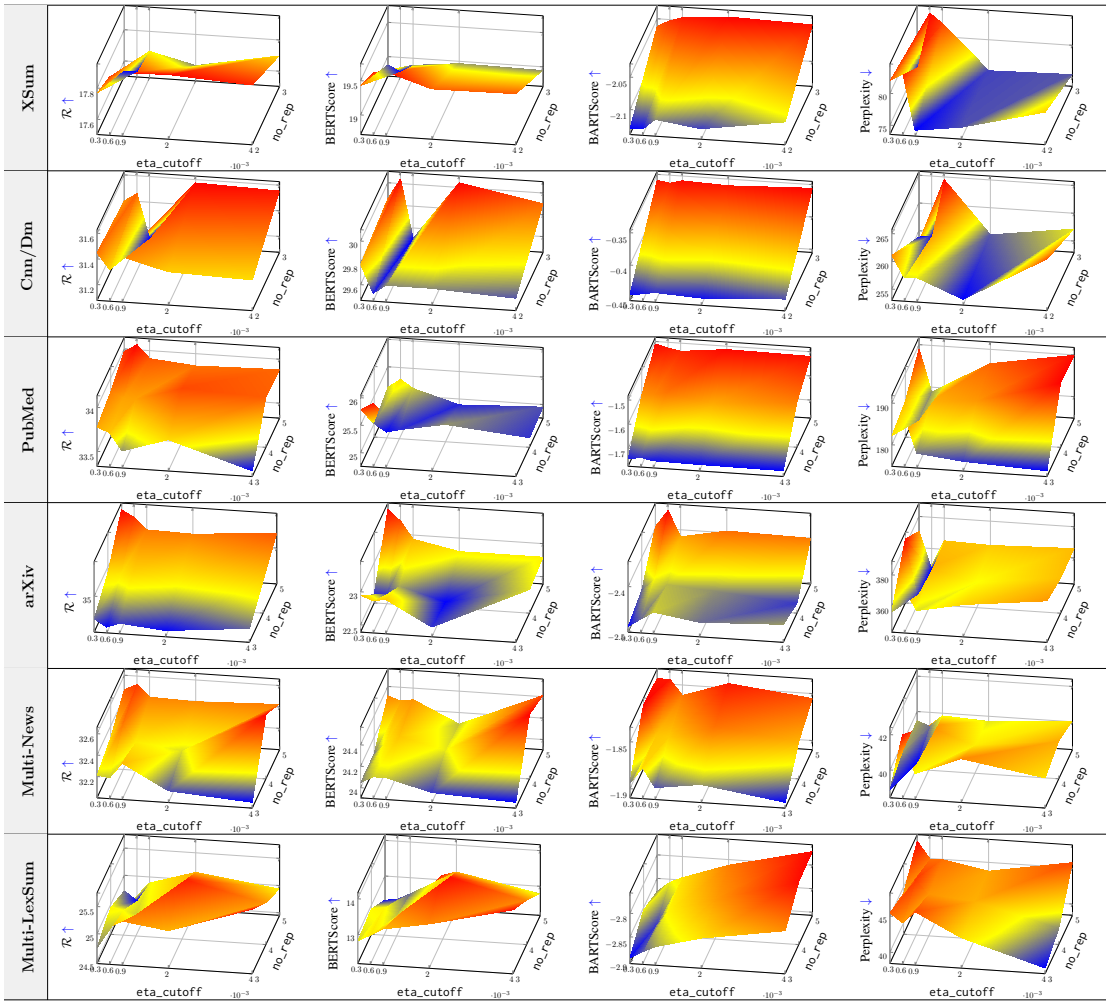
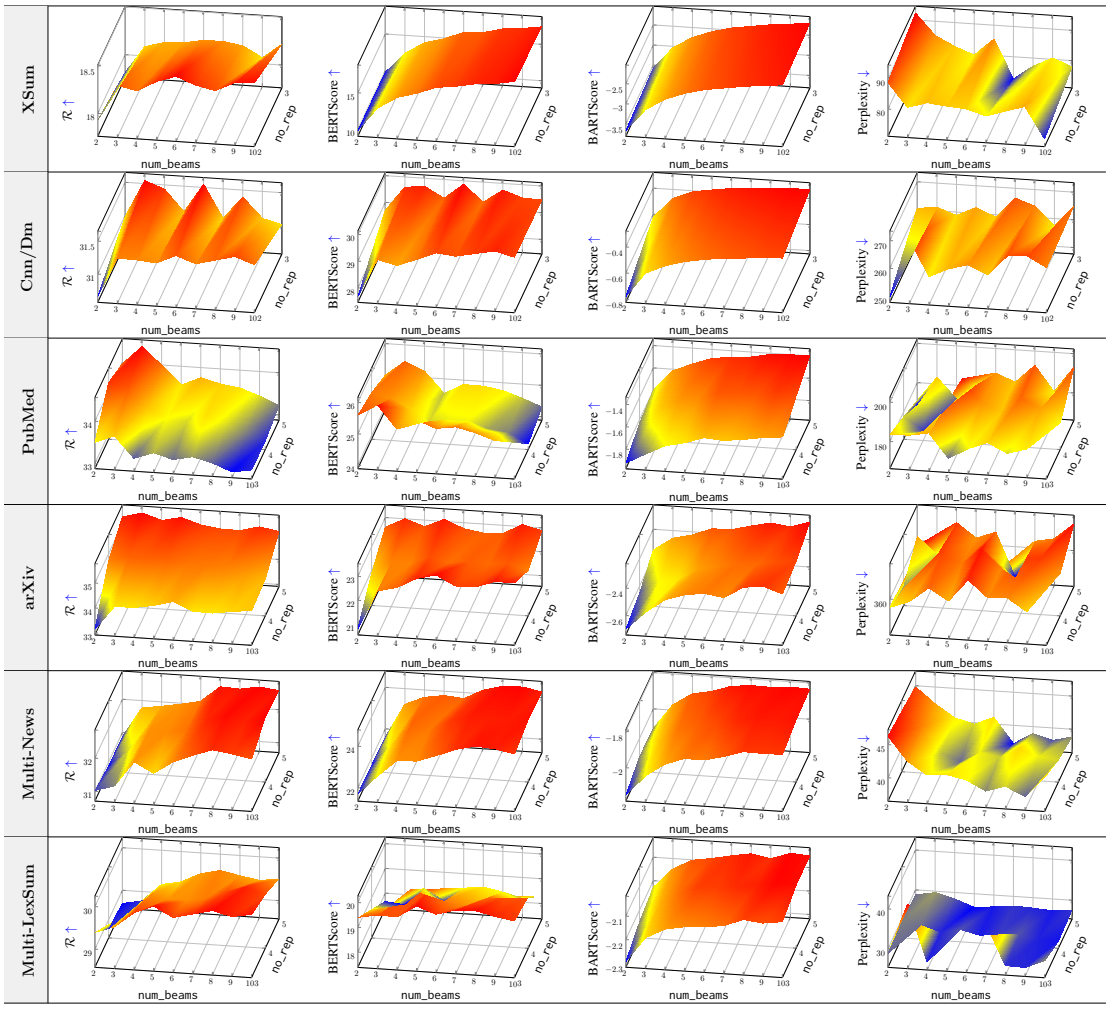


TABLE 6.28. Hyperparameters effect grouped by datasets – Beam Sampling.



6.3
Multi-Language Transfer Learning for Low-Resource Legal Case Summarization

Originally published as: *Gianluca Moro, Nicola Piscaglia, Luca Ragazzi, and Paolo Italiani. Multi-language transfer learning for low-resource legal case summarization. Artificial Intelligence and Law, pages 1–29, 2023a. doi: 10.1007/s10506-023-09373-8. URL https://doi.org/10.1007/s10506-023-09373-8*

Abstract

Analyzing and evaluating legal case reports are labor-intensive tasks for judges and lawyers, who usually base their decisions on report abstracts, legal principles, and commonsense reasoning. Therefore, summarizing legal documents is time-consuming and requires excellent human expertise. Moreover, public legal corpora of specific languages are almost unavailable. This paper proposes a transfer learning approach with extractive and abstractive techniques to cope with the lack of labeled legal summarization datasets, namely a low-resource scenario. In particular, we conducted extensive multi- and cross-language experiments. The proposed work outperforms the state-of-the-art results of extractive summarization on the Australian Legal Case Reports dataset and sets a new baseline for abstractive summarization. Finally, syntactic and semantic metrics assessments have been carried out to evaluate the accuracy and the factual consistency of machine-generated legal summaries.

6.3.1 Introduction

Automatic text summarization is undoubtedly one of the most valuable deep learning applications for natural language processing (NLP), especially when document analysis is time-consuming for humans. Reading and evaluating legal case reports is labor-intensive for judges and lawyers, who usually base their choices on report abstracts, legal principles, and commonsense reasoning. The abstracts of legal reports are not widely available, and the text summarization task requires the expertise of law experts and a long time to perform. Furthermore, few datasets are publicly ready for text summarization in the legal domain, mainly if we need corpora written in a specific language because legislation is drafted in its relative nation's language. Therefore, the challenge of this work is to build an automatic summarization system of legal reports to speed up human productivity and overcome the dearth of available legal corpora.

Two main approaches have been proposed in the literature concerning the text summarization task: extractive and abstractive. Extractive summarization aims to select the most salient sentences in a text, obtaining a summary with exact sentences from

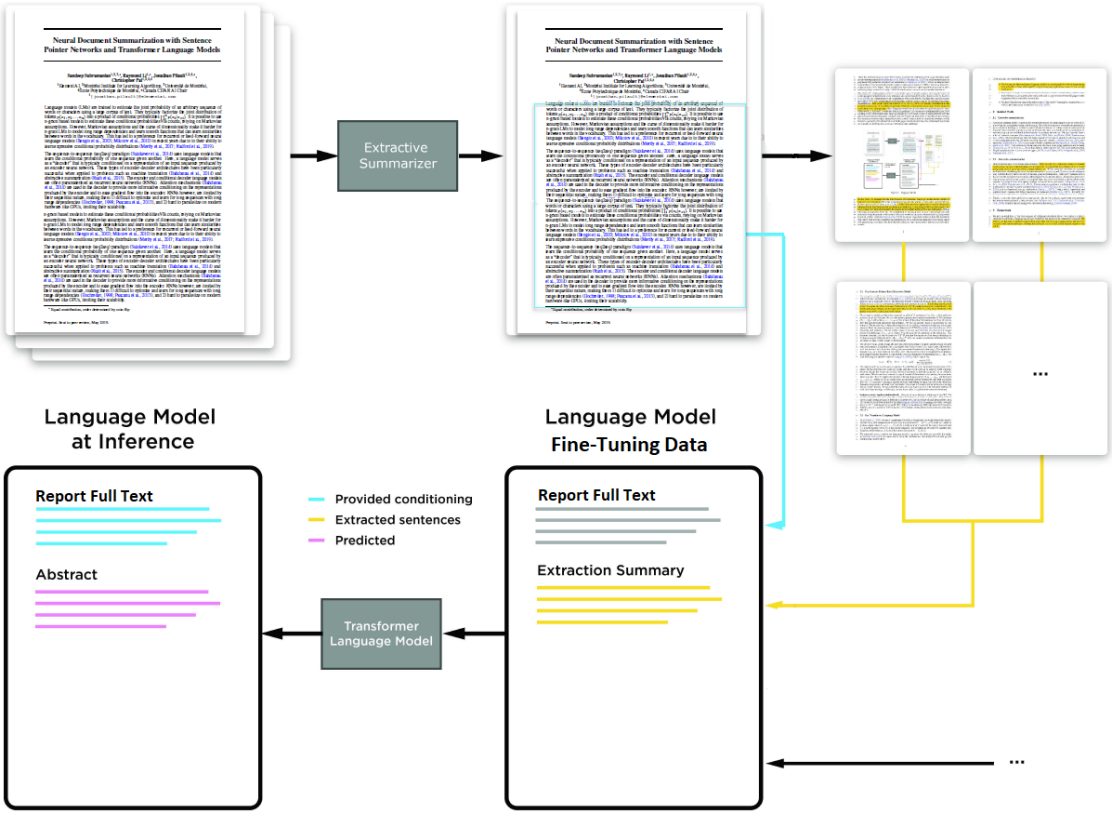


FIGURE 6.17. The overview of our solution for the abstractive summarization of a legal case report. First, our extractive summarizer (CNN+GRU) retrieves relevant sentences from the document. Next, these sentences are provided along with the whole legal case report to be arranged in the following order: (i) Report Full Text, (ii) Extracted Sentences. Then, a transformer language model (GPT-2 small) is trained on legal case reports organized in such a format. During inference, the full text of each new legal case report to summarize is given to the language model as a context to generate a summary.

the original document. Instead, the abstractive approach summarizes an input text by rephrasing it and also using new words that may not be present in the source text.

In this work, both extractive and abstractive summarization techniques have been tackled by proposing a transfer learning approach that can be used to cope with the lack of labeled legal summarization datasets (i.e., legal corpora without human-generated abstracts), which is a typical low-resource scenario concerning the available data. Our method allows the generation of abstractive summaries from tagged catchphrases within legal reports (Figure 6.17). The catchphrases are meant to present the essential legal concepts of a case. To this end, we first select and extract the relevant sentences in the text, if not already tagged, by using a lightweight neural model composed of CNN and GRU layers. Then, we pass the extracted sentences to GPT-2 [Radford et al., 2019a] as the reference summary. We chose GPT-2 because it is a generative decoder-only transformer-based model that is more efficient than the usual sequence-to-sequence summarization models based on encoder and decoder layers that double the memory

space and training time.

Our approach can be considered a general solution to overcome the absence of abstractive reference summaries written by human legal experts. Indeed, only catchphrase tags are required, which are much less time-consuming to create than human-written abstracts because they can be obtained in multiple ways, for example, by applying an unsupervised extractive algorithm (e.g., TextRank [Mihalcea and Tarau, 2004]) or manually tagging the sentences of a few documents and afterward fine-tuning a pre-trained model.

We experiment with the Australian Legal Case Reports dataset, evaluating our model's effectiveness in summarizing legal case reports in several languages (i.e., Australian, Italian, German, Spanish, Danish, French) whose translations are automatically generated with the Google Translate API as in prior works [Feng et al., 2022a]. We also test cross-language performance by directly summarizing English-written texts in all benchmarked languages. Finally, we assess the factual consistency [Kryscinski et al., 2020] of the abstractive summaries generated.

The paper is organized as follows. Section 6.3.2 analyzes the literature on text summarization and deepens the related works on the Australian Legal Case Reports dataset. Section 6.3.3 presents our transfer learning approach and the models employed for the summarization tasks. Section 6.3.4 shows the multi-language experiments with extractive and abstractive techniques. Lastly, Section 6.3.5 sums up the work with final thoughts.

6.3.2 Related Work

In order to achieve state-of-the-art (SOTA) results in automatic text summarization, many advances have been made in neural network architectures [Cho et al., 2014, Sutskever et al., 2014, Bahdanau et al., 2015, Vinyals et al., 2015, Vaswani et al., 2017, Moro et al., 2022a], pretraining and transfer learning [McCann et al., 2017, Peters et al., 2018, Devlin et al., 2019], and the availability of large-scale supervised datasets [Sandhaus, 2008, Nallapati et al., 2016, Grusky et al., 2018, Narayan et al., 2018, Sharma et al., 2019], which allowed deep learning-based approaches to dominate the field and overcome the need for low-resource summarization techniques [Moro and Ragazzi, 2022, Moro et al., 2023b,d, Moro and Ragazzi, 2023]. SOTA solutions leverage attention layers [Liu, 2019, Liu and Lapata, 2019b, Zhang et al., 2019], copying mechanisms [See et al., 2017, Cohan et al., 2018], and multi-objective training strategies [Guo et al., 2018, Pasunuru and Bansal, 2018], including reinforcement learning (RL) techniques [Kryscinski et al., 2018, Dong et al., 2018, Wu and Hu, 2018].

SOTA solutions for extractive summarization include BERT-based models, such as BertSum [Liu and Lapata, 2019b], BERT + RL [Bae et al., 2019], and PnBert [Zhong et al., 2019]. On the contrary, abstractive summarization includes transformer-based models for sequence-to-sequence learning, such as BART [Lewis et al., 2020a], PEGASUS [Zhang et al., 2020a], T5 [Raffel et al., 2020], and ProphetNet [Qi et al., 2020]. Furthermore, in the new research focused on building efficient transformers with linear complexity, the SOTA models in long document summarization are Longformer Encoder-

Decoder [Beltagy et al., 2020], BigBird-Pegasus [Zaheer et al., 2020], and Hepos [Huang et al., 2021].

The summarization task on the Australian Legal Case Reports dataset was first tackled by Galgani and Hoffmann [2010]. They presented a new knowledge-based approach to legal citation classification and created an extensive training and test corpus from court decision reports in Australia. Their later work [Galgani et al., 2012b] presents challenges and possibilities for the automatic generation of catchphrases for legal documents. The authors developed a corpus of human-generated legal catchphrases that they can compute statistics useful for automatic catchphrase extraction. Afterward, Galgani et al. [2012c] presented their approach to assigning categories and generating catchphrases for legal case reports. They describe their knowledge acquisition framework, which lets them quickly build classification rules, using a few features to assign general labels to cases. They show how the resulting knowledge base outperforms machine learning models, which use both the designed features or a traditional bag-of-words representation. In the same year, Galgani et al. [2012a] described a hybrid approach in which several different summarization techniques are combined in a rule-based system using manual knowledge acquisition. Here, human intuition, supported by data, specifies attributes and algorithms and the contexts where they are best used. Lastly, Galgani et al. [2012d] presented an approach to use incoming and outgoing citation information to automatically generate catchphrases for legal case reports. Specifically, they created a corpus of cases, catchphrases, and citations and performed a ROUGE-based evaluation [Lin, 2004], which showed the superiority of their citation-based methods over those of full text only.

Mandal et al. [2017] proposed an unsupervised approach for extracting and ranking catchphrases from the same court case documents focusing on noun phrases. They compared the proposed approach with several unsupervised and supervised baselines, showing that the proposed methodology achieves statistically significantly better performance than all baselines.

In the latest published work on Australian legal cases, Tran et al. [2018] presented a method of automatic catchphrase extraction. They utilized deep neural networks to construct a scoring model for their extraction system and achieved comparable performance without using citation information.

Other recent work focused on benchmarking different extractive and abstractive approaches [Shukla et al., 2022] without considering the extraction of catchphrases or experimenting with it in different datasets [Koboyatshwene et al., 2017, Bhargava et al., 2017, Kayalvizhi and Thenmozhi, 2020].

Our work focused on the Australian Legal Case Reports dataset for two main reasons. First, several studies have already been performed on such a dataset. Second, it is suitable for simulating the lack of human-crafted abstracts, unlike the BillSum dataset [Kornilova and Eidelman, 2019], for example. Furthermore, this work has been partially inspired by a recent approach proposed by Pilault et al. [2020], which combines extractive summarization with abstractive to increase the performance of their model. Nevertheless, since we do not have abstracts as reference summaries, we fine-tune a pre-trained

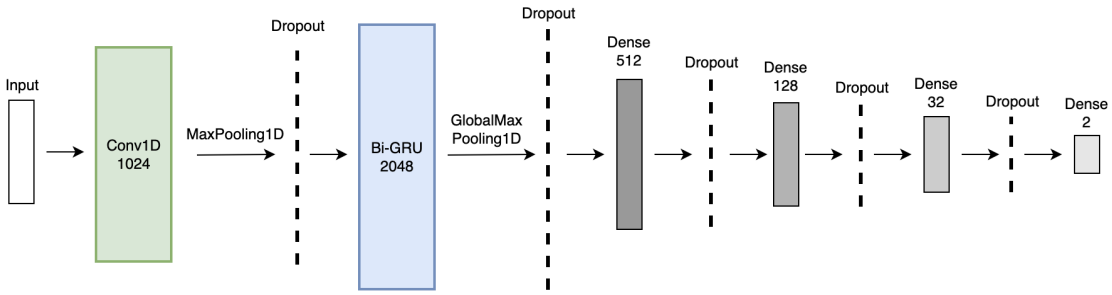


FIGURE 6.18. The model architecture used in the extractive summarization experiments.

transformer-based model using extractive summaries as reference ones to overcome that absence.

6.3.3 Method

In this work, we apply extractive and abstractive summarization techniques to propose a transfer learning approach to generate abstractive summaries starting from tagged catchphrases. In particular, BERT [Devlin et al., 2019] (base, multilingual cased) and a deep neural network composed of CNN and GRU layers have been used for the extraction phase, while the abstractive one has been performed with the GPT-2 transformer-based model.

6.3.3.1 Extractive Summarization

To generate contextualized word embeddings, we apply the BERT-Base-Multilingual-Cased pre-trained model for two main reasons: (i) BERT has achieved the SOTA in various NLP tasks, and (ii) the multilingual model allows us to overcome the absence of multi-lingual legal case datasets in the cross-language experiments. We obtained the sentence embeddings by performing the mean of the word embeddings related to words within the same sentence.

For the binary classification task, our model (Figure 6.18) consists of:

1. One 1D CNN layer [LeCun et al., 1999], with a subsequent MaxPooling1D operation, with a kernel of size 1 and filters of size 1024.

2. One bidirectional GRU layer [Cho et al., 2014] with GlobalMaxPooling1D operation.

3. Four fully connected layers of decreasing dimensionality.

All the main layers are interleaved with Dropout levels. Technically, the Conv1D, MaxPooling1D, and Dropout layers have only been applied when the mean of word embeddings has been used as the sentence embedding building method for performance reasons.

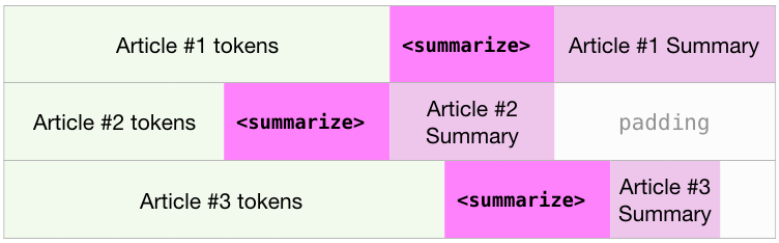


FIGURE 6.19. The structure of each instance of fine-tuning data used in the abstractive summarization with the GPT-2 language model. Each data is composed of: (i) the legal case report full text, (ii) a `<summarize>` tag, and (iii) the report summary obtained via the extractive summarization process.

Word/sentence embeddings have been produced using the Flair NLP library [Akbik et al., 2019].[19] This framework allows us to choose between different embedding methods and generate tensors of words/sentences from the relative string.

We train the extractive summarization model to minimize the categorical cross-entropy loss, as follows:

$$\mathcal{L}_{es} = -\sum_{i=1}^{i=N} y_i \cdot \log(\hat{y}_i) \tag{6.9}$$

where N is the number of samples and y_i and $\hat{y}_i$ are the gold and predicted labels, respectively.

6.3.3.2 Abstractive Summarization

GPT-2 has been used to generate legal abstractive summaries in the abstractive summarization scenario. To this end, two main steps are required:

1. Fine-tuning data (Figure 6.19): each instance of our GPT-2 fine-tuning data is sequentially composed of the **input text**, a **`<summarize>` tag**, and the **input text summary**. In our case, the input text will be the original text of each legal case report, whereas the summary will be the set of extracted sentences labeled as relevant for each report.

2. Inference phase (Figure 6.20): each test data instance is made up of the **original text** of a new legal report followed by the **`<summarize>` tag**.

The embedding process is integrated into the GPT-2 architecture, where the tokenizer transforms the sequences of words into numeric vectors.

We train the abstractive summarization model using the standard cross-entropy loss, which requires the model to predict the next token y_i of the target $\mathcal{Y}$ given $\mathcal{X}$ and the previous target tokens $y_{1:i-1}$, as follows:

$$\mathcal{L}_{as} = -\sum_{i=1}^{|\mathcal{Y}|} \log p(y_i | y_{1:i-1}, \mathcal{X}) \tag{6.10}$$

[19] https://github.com/flairNLP/flair.

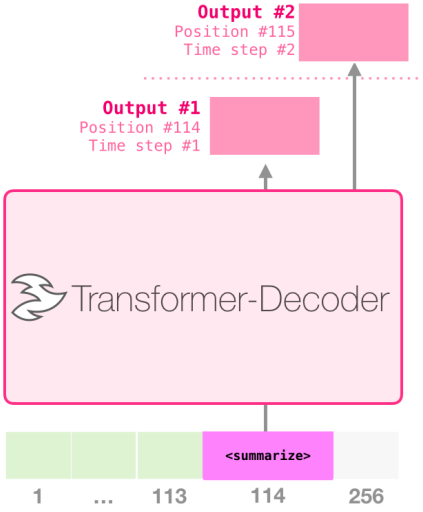


FIGURE 6.20. The inference phase in the abstractive summarization process performed via GPT-2. For each summary to produce, the full text of the legal case report is concatenated to the `<summarize>` tag. The GPT-2 model iterates a specified number of steps producing one token at a time. The report summary will be the tokens generated after the `<summarize>` tag.

where p is the predicted probability over the vocabulary.

6.3.4 Experiments

This section first introduces the dataset, experimental setup, and evaluation metrics. Then, we dive deeper into the in- and cross-domain experiments for extractive and abstractive approaches.

6.3.4.1 Dataset

The dataset used in our experiments is the Australian Legal Case Reports and represents a textual corpus of around 4000 legal cases for automatic summarization and citation analysis [Galgani et al., 2012d]. The dataset contains Australian legal cases from the Federal Court of Australia (FCA) from 2006 to 2009, downloaded from AustLII.[20] For each document, the authors collected catchphrases, citation sentences, citation catchphrases, and citation classes that indicate the type of treatment given to the cases cited in the present case.

The dataset is structured in three directories:

- **fulltext**: it contains the full text and the catchphrases of all the cases from the FCA. Each document (`<case>`) contains:
  - `<name>`: the name of the case.

[20] http://www.austlii.edu.au/.

  - `<AustLII>`: the link to the page from where the document was taken.
  - `<catchphrases>`: a list of `<catchphrase>` elements.
  - `<sentences>`: a list of `<sentence>` elements.

- **citations_summ**: it contains citations element for each case with the following fields:
  - `<name>`: the name of the case.
  - `<AustLII>`: the link to the page from which the document was taken.
  - `<citphrases>`: a list of `<citphrase>` elements that are catchphrases of a case that is cited or cite the current. The attributes are `id`, `type` (cited or citing), and `from` (the case from which the catchphrase is taken).
  - `<citances>`: a list of `<citance>` elements that are sentences from a later case that mention the current case. They also have the `from` attributes.
  - `<legistitles>`: a list of `<title>` elements that are titles of a piece of legislation cited by the current case.

- **citations_class**: it contains for each case a list of labeled citations with the following fields:
  - `<name>`: the name of the case.
  - `<AustLII>`: the link to the page from which the document was taken.
  - `<citations>`: a list of `<citation>` elements. They contain several attributes, such as the `<class>` of the citation as indicated in the document (`considered`, `followed`, `cited`, `applied`, `notfollowed`, `referred to`, etc.), the name of the case which is cited (`<tocase>`), the link to the document of the case which is cited (`<AustLII>`), and the `<text>` paragraphs in the case cited where the current case is mentioned.

No missing values have been found in `fulltext` and `citation_class` directory files, whereas some values are missing in `citation_summ` documents.

XML files contain many HTML entity characters. These latter ones made XML parsing invalid since "&" characters would indicate entities of XML types and not HTML ones, so they had to be removed.

The data used to perform the analysis have been selected from the `fulltext` directory. For each legal case (i.e., an XML file), `<name>`, `<catchphrase>`, and `<sentence>` have been used. Special HTML entities have been removed to parse the text correctly. To this end, we replace HTML entity characters with the corresponding textual representation. Some legal case reports have been truncated as they are not encoded as UTF-8 strings.

To create the target variable (i.e., the feature representing the class) of our extractive summarization experiments, a label is needed for each legal sentence, indicating whether a sentence should be included in a case report summary. Thus, the class variable will

TABLE 6.29. Statistics of the legal case reports translated into all the evaluation languages.

	Source		Target			
	# Words	# Sentences	# Words	# Sentences	% Catchphrases	Compression Ratio
100 legal case reports						
Australian	6597	251	590	15	6.69	38.62
Italian	6081	232	606	16	7.01	27.37
German	5522	216	450	11	6.58	38.42
Spanish	5346	195	459	11	6.55	33.98
Danish	4326	174	353	9	6.60	33.98
French	5602	206	462	11	6.59	38.94
1000 legal case reports						
Australian	7365	280	615	16	6.22	36.42
Italian	6869	267	579	15	6.57	37.74
German	6591	257	541	14	6.44	36.53
Spanish	7094	257	587	14	6.44	35.94
Danish	7777	311	639	17	6.44	37.21
French	6898	252	572	14	6.44	35.67

be binary. Since this information is not directly specified in the metadata, it has been generated using the following annotation process: for each sentence of a legal case, it is checked whether at least one of the catchphrases for that legal case is included in the current legal sentence examined. If this condition is true, then the label of that sentence will be `True`, else it will be `False`.

The legal sentences of each case report have been added to a common dataframe. So, each instance of this latter structure will represent a phrase. The instances in the dataframe have been balanced by class (represented by `is_catchphrase` attribute). Afterward, data instances were shuffled by groups of phrases from the same legal case report to avoid the situation where the model will classify phrases from legal case reports already seen during training time. Table 6.29 shows statistics for all datasets (i.e., the Australian Legal Case Reports translated into all evaluation languages), reporting the number of words and sentences in the source and target texts, the compression ratio between the source and the target (the number of source words divided by the number of target words), and the % of relevant sentences containing catchphrases.

6.3.4.2 Experimental Setup

Similar to previous contributions [Zhang et al., 2020a], all analyzes have been performed on a dataset of 100 and 1000 sizes, simulating real-world scenarios characterized by a lack of data. Precisely, one table has been generated for (i) each summarization technique adopted and (ii) each language in which the dataset has been translated. As commonly applied in realistic organizations due to the high cost of data labeling, all extractive and abstractive models have been trained using 70% of the dataset sampled in each experiment and tested in the remaining samples, similar to Bajaj et al. [2021].

Regarding the multi-lingual experiments, we translate each legal case report using

the Google Translate API in the following languages: Italian, German, Spanish, Danish, and French.

Implementation Details We fine-tuned the models based on Keras implementations for the tensor calculation, setting the seed to 7 for reproducibility. We trained the extractive summarization model for 100 epochs with a batch size of 50 and a learning rate of 1e-4, employing Adam as optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$ and a weight decay of 1e-6. We used the L1 and L2 regularization penalties of 0 and 0.001, respectively, and set the dropout at 0.1. Regarding abstractive summarization, we train the model for 4 epochs with a batch size of 4 and a learning rate of 5e-5, using Adam as the optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$ and a weight decay of 1e-6. For decoding, we used top-p nucleus sampling with top_p and temperature set to 0.9 and 1, respectively.

6.3.4.3 Evaluation Metrics

The extractive experiments have been tested by evaluating the F1 and ROUGE scores [Lin, 2004]. The F1 score is calculated between the sentences classified as relevant and the gold ones containing the catchphrases. F1 metric has been computed to consider both recall (i.e., the ratio of relevant sentences retrieved by the model out of all relevant sentences in the dataset) and precision metrics (i.e., the percentage of salient sentences retrieved out of all the sentences in the produced summary). Further, we choose the F1 metric because both the training and test set are not perfectly class-balanced since—after the class-balancing operation—data are grouped by case report groups and shuffled. By doing so, we also simulate a production environment in which an entirely new legal case report is passed as input to our classifier. On the contrary, the ROUGE scores are calculated between the concatenation of the classified sentences and the concatenation of the gold-relevant ones.

The evaluation metrics used for the abstractive summarization task are ROUGE and FactCC [Kryscinski et al., 2020]. The latter evaluates the factual consistency of the summaries w.r.t. the related report's original texts. In particular, the F1 score and balanced accuracy metrics have been calculated. Technically, we used the authors' data generation scripts to generate positive and negative examples from a JSONL file. Negative examples are created by applying some syntactic transformations to the original texts.

6.3.4.4 Extractive Summarization

Several extractive summarization experiments were carried out to evaluate the numerous embedding methods among those proposed in the literature (Table 6.30). BERT-Large is the best performer in creating embeddings of single words without the surrounding context, with 78.33% F1. The intuition behind the gap between BERT-Base-Multilingual and BERT-Large models is that the latter has been trained only on English text and has a larger dimension. However, we chose BERT-Base-Multilingual because it has

TABLE 6.30. The F1 scores obtained in the evaluation of several word embedding methods in the extractive summarization scenario. "Context" refers to using token embeddings' sequences as input to create contextual embeddings. The best scores are bolded.

	Word		**Context**	
Examples #	100	1000	100	1000
Models				
Word2Vec	67.10	72.69	**85.62**	87.83
GloVe	64.93	73.31	81.24	87.82
ELMo	**74.83**	75.89	79.69	88.35
BERT-Base-Multilingual	70.41	74.22	75.46	**90.29**
BERT-Large	70.92	**78.33**	83.92	85.22

similar performance and can be used as a baseline to compare the results in multi-lingual settings. On the other hand, models performed better where sequences as taken into account for creating contextualized word embeddings because orders are not lost with such a sentence representation, leading to higher F1 scores. In this case, BERT-Base-Multilingual is the best performer. Thus, the latter has been chosen, as it has the highest performance and can be used as a baseline to compare the results of experiments involving more than one language.

In-Domain Single-Language Experiments The detailed results of the in-domain extractive summarization tasks on the original Australian Legal Case Reports dataset are shown in Table 6.31. As expected, the ROUGE scores obtained in the experiments with 1000 legal case reports are the highest. These ROUGE scores outperform on average those produced by the latest work on the same dataset, which used a neural network for the catchphrase extraction task [Tran et al., 2018].

In-Domain Multi-Language Experiments Table 6.32 shows the results of the extractive summarization performed on 100 and 1000 legal case reports translated into Italian, German, Spanish, Danish, and French. Regarding 100 samples, the experiment with the Spanish dataset achieved the best results in almost all metrics except for the F1 score, which was obtained with the Danish dataset. Taking into account 1000 samples, the best performances were found in the German translation scenario except for the F1 score, which was achieved with the Italian dataset.

Figure 6.21 and Figure 6.22 sum up the results of the extractive experiments. Table 6.33 and Table 6.34 compare the multi-lingual results of other extractive summarization approaches, revealing the better performance of our solution. In detail, we compare with MemSum [Gu et al., 2022], BERT [Devlin et al., 2019], RoBERTa [Liu et al., 2019], and DistilBERT [Sanh et al., 2019]. For all languages, we use the corresponding multi-lingual model checkpoint (except for MemSum because it is available only for English and RoBERTa, which does not include Danish versions).

TABLE 6.31. The ROUGE (Precision, Recall, F1) and F1 scores obtained in the extractive summarization experiments using the original English-written reports and the two different strategies as the building method of sentence embeddings. The best scores are in bold.

Examples #	100			1000		
Mean of word embeddings						
F1 Score		70.41			**74.22**	
ROUGE-1	P: 42.01	R: 47.82	F1: 43.18	P: 47.99	R: 54.35	F1: **49.51**
ROUGE-2	P: 26.42	R: 30.04	F1: 26.94	P: 32.17	R: 35.57	F1: **32.91**
ROUGE-3	P: 23.24	R: 26.57	F1: 23.64	P: 28.79	R: 31.60	F1: **29.39**
ROUGE-4	P: 22.17	R: 25.44	F1: 22.53	P: 27.73	R: 30.44	F1: **28.28**
ROUGE-L	P: 44.62	R: 50.09	F1: 45.89	P: 49.79	R: 55.33	F1: **51.26**
ROUGE-w1.2	P: 31.02	R: 17.85	F1: 21.26	P: 34.91	R: 19.90	F1: **24.20**
Token embedding sequences						
F1 Score		75.46			**90.29**	
ROUGE-1	P: 60.27	R: 59.70	F1: 59.98	P: 57.36	R: 70.45	F1: **60.54**
ROUGE-2	P: 47.45	R: 51.55	F1: 48.53	P: 47.72	R: 56.62	F1: **50.07**
ROUGE-3	P: 45.17	R: 49.70	F1: 46.96	P: 45.40	R: 53.57	F1: **47.58**
ROUGE-4	P: 44.69	R: 49.35	F1: 46.56	P: 44.60	R: 52.64	F1: **46.73**
ROUGE-L	P: 62.41	R: 60.97	F1: 61.68	P: 59.90	R: 71.72	F1: **63.09**
ROUGE-w1.2	P: 45.82	R: 22.10	F1: 28.33	P: 41.73	R: 25.46	F1: **29.86**

TABLE 6.32. The ROUGE-F1 and F1 scores obtained in the extractive summarization experiments using 100 and 1000 Australian Legal Case Reports, including the original ones (written in English) and those translated into several languages (Italian, German, Spanish, Danish, French). The mean of word embeddings has been used as the building method for sentence embeddings. The best scores are bolded.

	F1	R-1	R-2	R-3	R-4	R-L	R-w1.2
100 legal case reports							
Australian	70.41	43.18	26.94	23.64	22.53	45.89	21.26
Italian	65.20	61.29	50.43	48.78	47.87	63.91	32.36
German	66.64	47.10	33.79	31.43	30.40	50.00	23.33
Spanish	69.68	**77.73**	**69.60**	**67.48**	**66.28**	**79.86**	**40.78**
Danish	**69.78**	60.69	50.15	48.00	46.95	63.44	32.16
French	67.97	59.85	46.82	44.11	42.82	61.36	29.34
1000 legal case reports							
Australian	74.22	49.51	32.91	29.39	28.28	51.26	24.20
Italian	**70.90**	64.74	55.68	53.83	52.79	67.69	34.68
German	69.62	**67.70**	**59.85**	**58.07**	**56.96**	**70.68**	**36.28**
Spanish	69.63	66.88	53.71	50.79	49.59	69.06	33.64
Danish	70.01	62.55	52.08	49.77	48.38	65.61	32.01
French	69.80	55.73	40.64	37.45	36.14	57.73	26.77

Cross-Language Experiments Table 6.35 shows the results of the extractive summarization task in the cross-domain scenario. The fine-tuning technique has been used as the transfer learning approach. The following experiments have been conducted:

- Regarding the experiments with 100 samples, (1) we tested the extractive model trained on 70 English reports directly on 30 cases of a different language, (2) we fine-tuned the extractive model (already trained on 70 English documents) with 70 reports of different languages and then tested it on 30 cases of a different language.
- About the experiments with 1000 samples, (1) we tested the model trained on 700 English samples directly on 30 cases of a different language, (2) we fine-tuned the model (trained on 700 English reports) with 70 documents of different languages and tested it on 30 cases of a different language.

Our models show a good generalization capability across different language domains. The application of the fine-tuning technique has boosted the considered evaluation metrics, showing how transfer learning can be used to overcome the lack of a labeled dataset in a specific language, as in the legal domain.

6.3.4.5 Abstractive Summarization

In-Domain Single-Language Experiments The results of the abstractive summarization tasks performed on the original Australian Legal Case Reports dataset are shown in Table 6.36. As expected, in the scenario with 1000 samples, the results are much higher for each metric than the ones obtained in the experiment with 100 reports. The results are similar to those obtained in the latest catchphrase extraction work proposed in the literature on the same dataset [Tran et al., 2018]. This gives us the intuition that our abstractive model can produce abstracts with a certain degree of lexical and syntactic correctness.

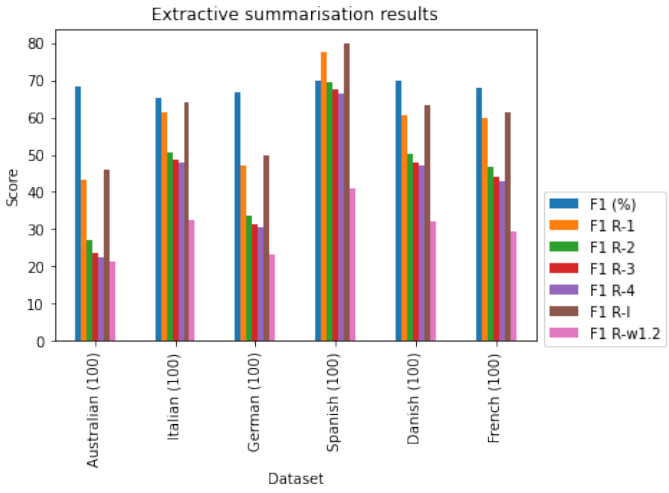


FIGURE 6.21. The ROUGE-F1 and F1% scores obtained in the extractive summarization experiments using 100 Australian legal case reports translated into multiple languages. The mean of word embeddings has been used as the building method for sentence embeddings.

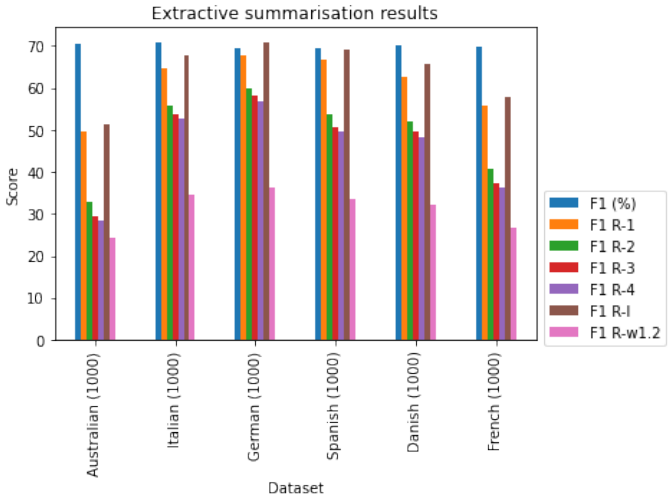


FIGURE 6.22. The ROUGE-F1 and F1% scores obtained in the extractive summarization experiments using 1000 Australian legal case reports translated into multiple languages. The mean of word embeddings has been used as the building method for sentence embeddings.

In-Domain Multi-Language Experiments Table 6.37 shows the results of the abstractive summarization performed on 100 and 1000 translated reports from the Australian Legal Case Reports dataset. Regarding 100 samples, the best ROUGE-3 score is achieved with the French dataset, while we obtained the best results from the experiment with the Spanish dataset. Considering 1000 documents, the best scores are achieved in the Spanish translation scenario for all metrics except ROUGE-3 and ROUGE-4, where the experiment with the French dataset got first place.

Figure 6.23 and Figure 6.24 sum up the results of the abstractive experiments.

FactCC Assessment Table 6.38 shows the results of the FactCC model evaluation in 60 and 1800 Australian legal case reports scenarios, respectively. The legal case report number is doubled since their approach provides transformations, which creates another dataset containing negative examples. As expected, the model fine-tuned with more examples (2100) achieves the best results among the two experiments, with a much higher balanced accuracy and F1 score. This makes us think that the model needs many training data to replicate or overcome the results obtained in the original paper [Kryscinski et al., 2020]. As the FactCC model has performed well on legal cases, it has been used as the metric for evaluating our generated abstractive summaries.

We aim to evaluate our abstractive summaries by running the previously trained FactCC model (with 2100 legal training reports). This latter has been used to classify 100 and 1000 abstractive machine-generated summaries as `CORRECT` or `INCORRECT`. If a summary is evaluated as `CORRECT`, we have reasonable assurance that it is fluently and coherently written. The training technique is BERT-base-uncased fine-tuning for 8 epochs with the default parameters of the FactCC model. Table 6.39 shows the ratio

TABLE 6.33. The comparison with extractive summarization models on the multi-language experiment with 100 labeled samples. The best scores are bolded.

100 samples	F1	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-L	ROUGE-w1.2
Australian							
MemSum	20.47	32.53	15.93	12.90	12.30	26.67	11.03
BERT	28.05	35.43	23.33	21.15	20.67	32.48	14.84
RoBERTa	30.17	35.78	24.07	21.93	21.40	32.72	15.19
DistilBERT	32.19	34.72	23.97	21.76	21.21	31.85	15.11
Our	**70.41**	**43.18**	**26.94**	**23.64**	**22.53**	**45.89**	**21.26**
Italian							
BERT	43.38	39.83	31.37	30.25	29.84	38.16	18.92
RoBERTa	44.55	38.80	31.04	29.91	29.43	37.77	18.72
DistilBERT	41.73	39.83	31.00	30.10	29.80	38.49	19.06
Our	**65.20**	**61.29**	**50.43**	**48.78**	**47.87**	**63.91**	**32.36**
German							
BERT	33.74	28.01	20.70	19.42	18.85	27.18	12.83
RoBERTa	18.96	18.67	15.32	14.84	14.68	18.22	9.21
DistilBERT	27.82	27.87	20.37	19.05	18.45	27.00	12.70
Our	**66.64**	**47.10**	**33.79**	**31.43**	**30.40**	**50.00**	**23.33**
Spanish							
BERT	24.61	30.46	19.13	17.14	16.33	27.54	12.04
RoBERTa	32.64	33.57	22.16	20.00	19.23	30.00	13.62
DistilBERT	28.47	30.87	19.98	17.84	17.07	27.85	12.33
Our	**69.68**	**77.73**	**69.60**	**67.48**	**66.28**	**79.86**	**40.78**
Danish							
BERT	28.58	28.63	19.57	18.23	17.62	25.99	11.81
DistilBERT	27.52	24.40	15.81	14.57	14.04	22.85	9.74
Our	**69.78**	**60.69**	**50.15**	**48.00**	**46.95**	**63.44**	**32.16**
French							
BERT	32.52	29.10	18.28	16.16	15.43	25.86	10.99
RoBERTa	25.50	32.96	21.62	19.47	18.69	28.86	13.05
DistilBERT	31.26	33.37	23.17	21.04	20.28	30.04	13.90
Our	**67.97**	**59.85**	**46.82**	**44.11**	**42.82**	**61.36**	**29.34**

of report summaries classified as `CORRECT` out of all evaluated summaries. In evaluating 650 abstractive summaries, almost 50% of them are classified as consistent.

Although the limits of the evaluation method are apparent, we believe that this result shows how GPT-2 can produce abstracts with a certain degree of consistency even in the legal domain. Furthermore, this evaluation represents a baseline that may be improved with future enhancements to the summarization technique.

6.3.4.6 Evaluation

Extractive Summarization In order to compare our results with the SOTA, we searched for all the works that used the Australian Legal Case Reports dataset. The

TABLE 6.34. The comparison with extractive summarization models on the multi-language experiment with 1000 labeled samples. The best scores are bolded.

1000 samples	F1	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-L	ROUGE-w1.2
Australian							
MemSum	28.58	34.94	23.17	20.65	19.76	30.79	13.68
BERT	46.50	42.74	**34.39**	32.77	32.32	40.53	20.66
RoBERTa	46.02	39.83	34.21	**33.22**	**32.93**	38.65	20.36
DistilBERT	46.92	41.60	33.61	32.13	31.70	39.63	20.21
Our	**74.22**	**49.51**	32.91	29.39	28.28	**51.26**	**24.20**
Italian							
BERT	44.85	35.70	29.27	28.38	28.02	34.20	17.55
RoBERTa	42.67	34.46	28.03	27.18	26.81	33.11	16.94
DistilBERT	44.75	35.63	29.22	28.39	28.04	34.21	17.51
Our	**70.90**	**64.74**	**55.68**	**53.83**	**52.79**	**67.69**	**34.68**
German							
BERT	43.80	33.01	27.54	26.80	26.52	32.02	16.48
RoBERTa	23.14	26.90	21.08	20.21	19.86	25.82	12.82
DistilBERT	44.88	33.65	28.01	27.27	26.99	32.53	16.72
Our	**69.62**	**67.70**	**59.85**	**58.07**	**56.96**	**70.68**	**36.28**
Spanish							
BERT	47.40	37.51	30.64	29.45	29.07	36.15	18.25
RoBERTa	47.68	39.82	31.64	30.27	29.80	37.79	19.07
DistilBERT	47.02	39.13	30.97	29.48	28.99	36.82	18.50
Our	**69.63**	**66.88**	**53.71**	**50.79**	**49.59**	**69.06**	**33.64**
Danish							
BERT	28.15	20.89	18.33	18.05	17.90	20.60	10.75
DistilBERT	39.45	29.22	23.89	23.27	23.00	27.99	14.23
Our	**70.01**	**62.55**	**52.08**	**49.77**	**48.38**	**65.61**	**32.01**
French							
BERT	43.03	34.81	28.15	27.11	26.76	32.81	16.55
RoBERTa	35.69	33.06	25.92	24.63	24.13	30.83	15.14
DistilBERT	45.56	38.53	30.89	29.67	29.24	36.29	18.19
Our	**69.80**	**55.73**	**40.64**	**37.45**	**36.14**	**57.73**	**26.77**

latest one we found is Tran et al. [2018] (Table 6.40), and it has been used as the baseline for the extractive summarization tasks because it has analogies with our work:

- It used the same data.
- It did not involve citation data in the training process.
- It only used sentences and words from catchphrases as target data.

By comparing our results, it can be stated that our model achieved excellent performance in syntactic terms. In particular, the scores of ROUGE-1 and ROUGE-W.1.2 obtained in our experiments are much higher. The primary motivation could be to use BERT as the word/sentence embedding builder system. As explained in the first

Table 6.35. The ROUGE (F1) and F1 scores of the cross-language experiments, where our extractive model trained on English reports has been applied in two experiments: (i) it has been tested without fine-tuning on different languages (EN $\longrightarrow$ LN); (ii) it has been fine-tuned using the translated legal case reports and then tested (EN, LN $\longrightarrow$ LN). The mean of word embeddings has been used as the building method for sentence embeddings.

LN	F1	R-1	R-2	R-3	R-4	R-L	R-w1.2
70 EN $\longrightarrow$ 30 LN							
Italian	64.43	57.49	44.37	42.11	40.85	60.93	29.67
German	68.63	53.30	42.93	41.06	40.12	56.35	28.02
Spanish	64.88	51.58	36.65	33.64	32.51	54.24	25.54
Danish	64.81	51.78	40.00	37.68	36.40	54.96	26.25
French	71.28	63.61	53.10	50.71	49.46	66.24	33.53
70 EN, 70 LN $\longrightarrow$ 30 LN							
Italian	83.07	64.61	53.12	50.93	49.90	66.47	34.25
German	79.25	50.45	37.72	35.63	34.78	53.40	26.33
Spanish	85.32	65.71	54.65	52.91	52.13	68.07	34.33
Danish	78.33	54.78	45.58	43.86	42.85	58.09	28.15
French	86.99	67.09	57.64	55.58	54.61	69.29	35.32
700 EN $\longrightarrow$ 30 LN							
Italian	56.65	53.33	39.22	36.66	35.76	56.36	28.12
German	57.38	31.67	18.04	16.34	15.69	34.80	15.01
Spanish	64.52	45.70	28.28	25.25	24.39	47.68	21.38
Danish	60.06	32.33	18.57	16.85	15.93	34.85	14.90
French	62.82	48.33	34.16	31.86	30.91	50.83	23.71
700 EN, 70 LN $\longrightarrow$ 30 LN							
Italian	83.27	63.05	52.00	49.55	48.54	65.56	33.15
German	90.32	51.44	39.79	37.99	37.07	54.61	27.24
Spanish	86.29	64.58	53.09	50.91	49.87	66.99	33.27
Danish	85.70	45.27	31.45	29.66	28.63	47.81	22.13
French	80.00	62.08	51.72	49.72	48.65	64.61	32.16

Table 6.36. The ROUGE (Precision, Recall, F1) obtained in the abstractive summarization using the original English-written reports. The best scores are in bold.

Examples #	100			1000		
ROUGE-1	P: 26.55	R: 21.51	F1: 22.84	P: 28.93	R: 26.18	F1: **27.18**
ROUGE-2	P: 3.30	R: 2.76	F1: 2.88	P: 4.86	R: 4.43	F1: **4.59**
ROUGE-3	P: 0.17	R: 0.14	F1: 0.15	P: 0.99	R: 0.92	F1: **0.94**
ROUGE-4	P: 0.00	R: 0.00	F1: 0.00	P: 0.42	R: 0.41	F1: **0.42**
ROUGE-L	P: 28.10	R: 23.29	F1: 24.70	P: 30.37	R: 27.93	F1: **28.87**
ROUGE-w1.2	P: 14.74	R: 5.71	F1: 7.93	P: 15.68	R: 7.06	F1: **9.64**

chapter, choosing a good contextualized embedding model is crucial for better performance. Another reason could be the use of a more expressive classification model: we

TABLE 6.37. The ROUGE-F1 scores obtained in the abstractive summarization experiments using 100 and 1000 Australian Legal Case Reports, including the original ones (written in English) and those translated into several languages (Italian, German, Spanish, Danish, French). The best scores are bolded.

	R-1	R-2	R-3	R-4	R-L	R-w1.2
100 legal case reports						
Australian	22.84	2.88	0.15	0.00	24.70	7.93
Italian	22.00	1.19	0.10	0.00	24.25	8.06
German	21.83	2.51	0.41	0.03	24.12	8.02
Spanish	**33.47**	**4.93**	0.61	**0.19**	**36.31**	**11.97**
Danish	22.89	1.67	0.31	0.03	25.85	8.45
French	30.18	4.15	**0.78**	0.18	31.42	10.02
1000 legal case reports						
Australian	27.18	4.59	0.94	0.42	28.87	9.64
Italian	25.11	2.75	0.56	0.15	27.59	9.25
German	25.31	4.02	1.01	0.26	28.06	9.47
Spanish	**35.70**	**6.57**	1.56	0.53	**37.52**	**12.38**
Danish	25.39	2.83	0.73	0.23	27.90	9.15
French	32.44	6.14	**1.83**	**0.76**	33.46	10.97

used LSTM networks combined with CNN ones, whereas, in their work, only CNNs have been applied. Extractive summarization of translated reports achieves better results for almost all metrics than the English scenario. In particular, the best scores have been obtained by the experiment with the Spanish dataset for the ROUGE metrics and the Danish dataset for the F1 score. This gives us the intuition that the model benefits

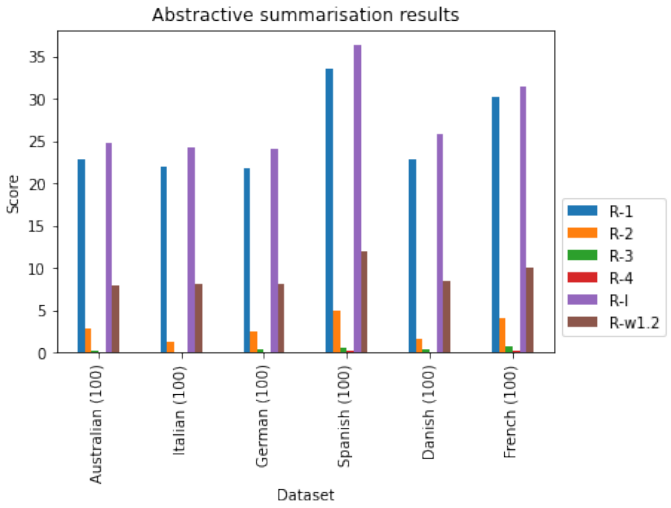


FIGURE 6.23. The ROUGE-F1 and F1% Scores obtained in the abstractive summarization experiments using 100 Australian legal case reports translated into multiple languages.

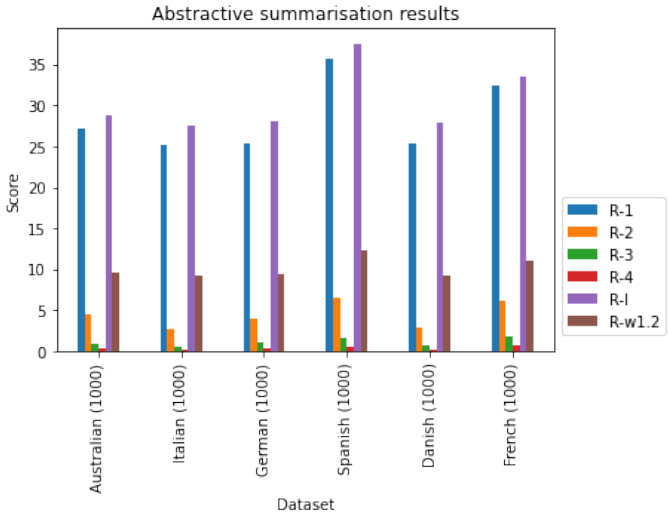


FIGURE 6.24. The ROUGE-F1 and F1% Scores obtained in the abstractive summarization experiments using 1000 Australian legal case reports translated into multiple languages.

from using the BERT multilingual model as the embedding builder. The latter lets us generate expressive contextualized word embeddings and allows us to work with different languages. In Table 6.41, we present a representative qualitative instance for each of the languages analyzed so far. The efficacy of our solution in extracting sentences that include catchphrases is readily apparent.

Abstractive Summarization Since no similar works have been found for abstractive summarization in the same dataset, extractive summarization results have been used as a baseline. As expected, the ROUGE scores of the abstractive summarization tasks are much lower than the extractive summarization ones. Indeed, the ROUGE score is a mere lexical measure. In the extractive scenario, if the model succeeds in classifying one sentence as relevant or not, then all the words of that sentence represent an overlapping between the generated and the reference summary. This does not apply to the abstractive summarization task by definition because here the goal is to produce a new summary, also using words that do not exist in the input text to summarize. However, the ROUGE scores of the abstractive summarization experiments are similar to Tran et al. [2018],

TABLE 6.38. The balanced accuracy and F1 scores of the FactCC model assessment after the fine-tuning using the Australian Legal Case Reports dataset (sampling 100 and 3000 reports, respectively).

Examples #	60	1800
Balanced Accuracy (%)	50.00	76.91
F1 Score (%)	66.15	77.50

TABLE 6.39. The percentage of inferred abstractive summaries classified as `CORRECT` (i.e., consistent with the related original full text) by our fine-tuned FactCC model in 3 experiments with different reports sampling (30, 300, and 650 machine-generated summaries via abstraction).

Predictions #	30	300	650
CORRECT ratio (%)	41.67	46.70	**49.92**

so we have the intuition that our abstractive model has been able to generate speeches, which are inherent to the input text, even if consistency and fluidity of speech are not yet verified. To do that, the FactCC model has been applied and it turned out that about 46% of 300 machine-generated reports have been classified as `CORRECT`. Even though our fine-tuned FactCC model has a 77% F1 score (i.e., it is affected by errors), this fact gives us the intuition that our abstractive summaries have a certain degree of fluency and coherency w.r.t. their related legal report original texts, which are not reflected in the ROUGE rating. Abstractive summarization of translated reports achieves similar results to the English scenario and is even better for the Spanish and French languages. This gives us the intuition that the model is still working well in languages other than English and could be applied to many use cases.

6.3.5 Conclusion

In this work, we tackle the automatic summarization of Australian legal case reports by presenting extractive and abstractive techniques. The abstractive solution can be considered a general approach to generating summaries despite the lack of human-crafted references. Our method only requires catchphrase tags that can be obtained in several ways: (i) by applying an unsupervised extractive summarization algorithm or (ii) by manually tagging the sentences of a few documents and afterward fine-tuning a pre-trained model.

We showed that our extractive summarization results overtake those produced by the latest work on the same dataset in the literature [Tran et al., 2018]. Instead, our abstractive summarization results led to ROUGE scores similar to theirs. In addition, we proved the speech consistency of our abstractive summaries using the FactCC model. Precisely, even though our fine-tuned FactCC model can make inference errors on test data (77% F1), the results suggest that our abstractive summaries have a certain degree of fluency and coherency w.r.t. their related legal sources, which are not reflected in the ROUGE rating.

TABLE 6.40. The ROUGE-1 and ROUGE-W-1.2 (Precision, Recall, F1) scores from "Automatic Catchphrase Extraction from Legal Case Documents via Scoring using Deep Neural Networks" by Tran et al. [2018].

ROUGE-1	P: 23.11	R: 30.84	F1: 22.95
ROUGE-w1.2	P: 14.50	R: 13.63	F1: 11.75

TABLE 6.41. Qualitative examples of extractive summarization in the multilingual setting. Catchphrases are colored yellow.

Language	Input Document	Extractive Summary
Australian	The circumstances revealed in this application for preliminary discovery suggest... ...Especially is this so where the person concerned is an Australian citizen and the information...	The circumstances revealed in this application for preliminary discovery suggest...
Italian	Il sig. Gormly sostiene che la negazione della giustizia naturale "contagia" l'intera seconda decisione del Tribunale... .... Il sig. Johnson, che si presenta per il ministro, sostiene che la base alternativa per il...	Il sig. Gormly sostiene che la negazione della giustizia naturale "contagia" l'intera...
German	Der vorliegende Fall wird von zwei Behörden in Bezug auf verstorbene Personen und Antidiskriminierungsgesetze sorgfältig eingegrenzt... ... In Stephenson gegen Human Rights and Equal Opportunity Commission (1996)..	Der vorliegende Fall wird von zwei Behörden in Bezug auf verstorbene Personen und Antidiskriminierungsgesetze sorgfältig eingegrenzt...
Spanish	...En Inglaterra, la prueba pericial se restringe a la que se requiere razonablemente para resolver los procedimientos. La licencia se concede o deniega como parte de la gestión del caso...	...En Inglaterra, la prueba pericial se restringe a la que se requiere razonablemente para resolver los procedimientos...
Danish	Dette er en ansogning om udeblivelsesdom i henhold til ordre 35A i Federal Court Rules... ... Den blev serveret fredag den 3. februar 2006, hvilket er mindre end de tre fridage...	Dette er en ansogning om udeblivelsesdom i henhold til ordre 35A i Federal Court Rules...
French	...Les membres du groupe de demande d'indemnisation sont presque exclusivement Yankunytjatjara ou Pitjantjatjara... ...Les requérants admettent que l'un des cinq requérants restants, Mantatjara Wilson, ne satisfait pas...	Les membres du groupe de demande d'indemnisation sont presque exclusivement Yankunytjatjara ou Pitjantjatjara...

Moreover, a translation task has been achieved to train our models and evaluate their ability to understand and summarize texts in several languages other than English. It turned out that the summarization of translated reports achieves better results than the English report scenario for some other languages. In particular, Spanish and French seem to perform generally better in the abstractive summarization case. In contrast, the summarization of German reports achieves the best results in the extractive summarization scenario with 1000 reports. Hence, our models summarize several languages effectively and could be applied to other legal case reports. Such experiments are supported by Google Translate API and the BERT multilingual embedding model (only used in the extractive summarization scenario). Finally, it turned out that our models can also generalize in a cross-language scenario, summarizing English reports directly in all the different languages.

The main challenges will be improving the quality of the machine-generated abstractive summaries and their evaluation. Some automatic evaluations like Kryscinski et al. [2020] have been proposed in the literature and represent an improvement over previous solutions to this problem, although they still have limitations.

Future works will be related to replicating the cross-language experiments using token embedding sequences as input instead of applying the mean of word embeddings to build sentence embeddings since the first method has led to higher results in the in-domain experiments with original Australian legal reports. Furthermore, the methods proposed in this work could be expanded by adding more advanced data techniques to FactCC to improve abstractive summaries evaluation and repeat the experiments using other SOTA large language models (e.g., ChatGPT) to improve the abstraction quality.

6.4
COMMA: A Multi-Task and Multi-Lingual Dataset of Constitutional Verdicts

Under review as: *Luca Ragazzi, Giacomo Frisoni, Gianluca Moro, Paolo Italiani, Lorenzo Molfetta, and Veronika Folin.* COMMA*: A multi-task and multi-lingual dataset of constitutional verdicts, 2024. Submitted to Data-centric Machine Learning Research*

Abstract

Transformer-based language models have sparked a revolutionary change in Legal NLP, endowing lawyers with unparalleled tools to effectively navigate, understand, and draft large volumes of text. However, the dearth of large-scale datasets from authoritative sources hampers further progress. The available resources are primarily single-task, English-only, and written in layman's terms. To bridge this gap, we introduce COMMA, a multi-task and multi-lingual archive of 14K verdicts drawn from the Constitutional Court of the Italian Republic, grounded in a non-common law system. Documents in COMMA diverge from ordinary legal manuscripts as they address fundamental principles and rights, entail technical jargon, exhibit articulated structure, are diachronic, have extended length, and demand more significant expertise and interpretation. By embracing 4 widespread languages, COMMA tackles a panoply of necessity-driven tasks: multi-granular abstractive summarization, decision generation, ruling classification, and articles retrieval. We systematically benchmark a catalog of language models in both few-shot and full settings, uncovering substantial headroom for improvement. We contribute to the new era of Legal NLP systems by openly releasing COMMA and best-performing models.

6.4.1 Introduction

Law is one of the most impactful domains in shaping our world, guiding every aspect of society, from safeguarding individual rights and ensuring public health to fostering social cohesion and resolving disputes. In this realm, written materials, such as legislation, judgments, and contracts, are ubiquitous and laden with jargon, formalities, and cross-references [Kornilova and Eidelman, 2019]. Consequently, NLP systems have the undisputed potential to maximize efficiency in legal professionals, freeing them from the shackles of repetitive, time-intensive, and knowledge-demanding tasks [Kanapala et al., 2019]. Transformer-based neural networks have revolutionized the AI landscape and have served as an affordable form of legal aid [Vold and Conrad, 2021]. With the increasing magnitude of language models (LMs) [Zhao et al., 2023], the necessity of arduous, real-world, and highly comprehensive datasets to coordinate scientific advances is also gaining momentum [Koch et al., 2021]. As detailed in Section 6.4.2, despite the

abundance of resources that have emerged over time, current legal corpora exhibit at least one of the following limitations: (i) they are not tailored for supervised training; (ii) they are crafted to tackle individual NLP tasks; (iii) they contain silver annotations; (iv) when geared toward generative applications, the targets are formulated in lay terms, intended for the general public instead of law practitioners; (v) they are predominantly devoted to English texts from US sources; (vi) they overlook the complete structure and metadata of the involved documents.

In this paper, we study the LM capabilities in a critical setting found in the Constitutional Court of the Italian Republic (CCIR), the highest judicial body in Italy. The CCIR receives a substantial influx of motions that exceed the threshold of 300 annually.[21] Intending to protect constitutional integrity, its decisions have a far-reaching impact, charting the course of legal precedents and doctrinal evolution. Reaching a consensus and writing a ruling are complex and laborious cognitive activities that require judges to search, link, interpret, apply, synthesize, and categorize multiple documents of different types. Streamlining and automating these multifaceted tasks would empower the decision-making process of the CCIR, while also improving the understanding of disputes and their resolutions.

To achieve this goal and address the shortcomings of current legal datasets, we openly release COMMA,[22] a constitutional court multi-task and multi-lingual archive consisting of 14K CCIR rulings with annotations authored by experts. In particular, COMMA embodies unique characteristics that make it a valuable study object for broader NLP research.

- COMMA forms a **diachronic collection of lengthy structured documents** (3418 English words, on average) spanning 66 years of judicial history and a wide spectrum of topics.
- In contrast to aggregated benchmarks with pre-existing datasets, e.g., Chalkidis et al. [2022] and Niklaus et al. [2023a], COMMA is entirely new and includes **7 necessity-driven tasks belonging to generation, classification, and retrieval families**.
- COMMA breaks down geographical barriers by presenting rulings, constitutional parameters, and task-specific targets in **4 languages**, incorporating semi-automatic translations in English, Spanish, and French.
- To the best of our knowledge, COMMA is **the first multi-task and multi-lingual benchmark that does not follow a common law system**. Unlike previous contributions [Chalkidis et al., 2022, Guha et al., 2022], verdicts are code-based and do not originate under legal precedents (i.e., past court decisions only play

[21] https://bit.ly/the-italian-constitutional-court

[22] The term "comma" has a Greek etymology (*κόμμα*), denoting a brief pause or interruption in speech or text. In the context of Italian law, it means a component of an article, regulation, or other normative act.

supplementary roles). Thus, NLP models must learn to reason on top of explicit rules, grasping their intended scope, exceptions, and ambiguities.

We drive exhaustive quantitative and qualitative experiments on COMMA, applying a mosaic of metrics and laying the foundation for future methodological developments. Overall, head-to-head evaluations of three cutting-edge LMs in few-shot and full settings reveal the need for substantial improvements for effective real-world use.

6.4.2 Related Work

6.4.2.1 Legal Language Models

From a holistic view, AI & Law is a long-standing field with publications dating back many decades [Governatori et al., 2022]. Bibliometric statistics linked to the top-tier NLP and AI venues exemplify its remarkable surge, with 600+ works screened between 2013 and 2022 [Katz et al., 2023]. Legal text analysis has been reshaped from 2017 onward based on transformer-based architectures in the neural era. Still, lawsuits present a non-trivial challenge in training machines to understand legal language accurately. Compared to general models, legal LMs often benefit from ad hoc pretraining [Chalkidis et al., 2020], currently achieving state-of-the-art results on various benchmarks [Zheng et al., 2021, Chalkidis et al., 2022, 2023, Niklaus et al., 2023a]. However, unlike encoder-only networks, the scarcity of public generative models pretrained on legal corpora persists, compelling the utilization of non-specialized alternatives [Hwang et al., 2022, Shen et al., 2022]. Despite some criticisms [Bender et al., 2021], the latest waves of large language models (LLMs) have delivered remarkable classification results through few-shot in-context exemplars [Guha et al., 2022, Sun, 2023]. On the other hand, there are strong reasons to believe that general NLP solutions will likely not eclipse equally sized law-specific networks. Open-source legal-based LLMs are restricted to Chinese [Song et al., 2023] and have extensive training hardware requirements. Furthermore, the distillation of LLM is a false promise for tasks not strongly supported in imitation data [Gudibande et al., 2023]. An in-depth literature analysis [Katz et al., 2023] shows that English is by far the most popular Legal NLP language (56%), trailed by Chinese (∼10%). When viewing models through the lens of low resources [Moro et al., 2023a], the diversity of languages has been notable over time, encompassing Italian [Bellandi et al., 2022, Galli et al., 2022, Licari and Comandé, 2022, Tagarelli and Simeri, 2022]. Taking COMMA as a testbed, we fairly compare the effectiveness and efficiency of available multi-lingual models on commodity hardware, examining their adaptability to different tasks, languages, and amounts of labeled training data.

6.4.2.2 Datasets for Legal Language Models

Law-related workstreams are typically decomposed into arrays of constituent tasks. Lamentably, despite being meaningful indicators of scientific progress, legal datasets are scattered and over-focused on single tasks. Classification, at both the sentence and document levels, is prevalent, ranging from case categories [Hwang et al., 2022]

TABLE 6.42. Summary of Legal NLP benchmarks.

Name	Novelty*	Task Families (Number)**	Datasets	Languages	Jurisdiction
LEXGLUE [Chalkidis et al., 2022]	✗	MLC (x4), SLC (x2), QA (x1)	6	English	EU,¶ US
LBOX [Hwang et al., 2022]	✓	SLC (x5), R (x2), AS (x1)	5	Korean	Korea¶
LEGALBENCH [Guha et al., 2022]	✓	SLC (x43), QA (x1)†	8	English	US
LEXTREME [Niklaus et al., 2023a]	✗	SLC (x7), MLC (x6), NER (x5)	11	24‡	Brazil,¶ Germany,¶ Switzerland,¶ Greece,¶ EU,¶ Romania¶
COMMA *(Ours)*	✓	AS (x3), SLC (x2), IR (x1), TG (x1)	1	Italian, English, Spanish, French	Italy¶

* ✓ = the benchmark does not consist of pre-existing datasets, ✗ = otherwise.
** *AS* = Abstractive Summarization, *TG* = Text Generation, *IR* = Information Retrieval, *MLC* = Multi-Label Classification, *NER* = Named Entity Recognition, *QA* = Question Answering, *R* = Regression, *SLC* = Single-Label Classification.
† LLM evaluation. Only $[0, 5]$ training instances and $[50, 100]$ test instances per task.
‡ The number of languages depends on the dataset (24 is limited to 2 NER datasets). Median = 1.
¶ Civil law systems.

and acceptance degrees [Lage-Freitas et al., 2022] to topics [Tuggener et al., 2020] and argumentative functions [Urchs et al., 2021]. Information retrieval (e.g., case matching) [Bhattacharya et al., 2019, Chalkidis et al., 2021b, Sansone and Sperlí, 2022] and extraction [de Gibert Bonet et al., 2022] immediately follow. Compared to these activities, scientific production orbiting natural language generation has been significantly lower over the past ten years (∼30x less) [Katz et al., 2023]. In this context, summarization datasets cover mainly UK judgments [Grover et al., 2004], US bills [Kornilova and Eidelman, 2019], US patents [Sharma et al., 2019], US government reports [Huang et al., 2021], and US civil rights [Shen et al., 2022]—frequently employing linguistically-simplified targets [Kornilova and Eidelman, 2019]. Supervised corpora are also available for automatic translation of European Parliament debates [Koehn, 2005, Iranzo-Sánchez et al., 2020]. More related to us, multi-task benchmarks gauge law mastery across various dimensions (Table 6.42). However, they prioritize simple classification tasks, mono-lingual texts [Chalkidis et al., 2022, Guha et al., 2022, Hwang et al., 2022], few labels [Niklaus et al., 2023a], and already published datasets [Chalkidis et al., 2022, Niklaus et al., 2023a]. Raw English document collections have recently been released for large-scale self-supervised pretraining [Henderson et al., 2022]. Instead, COMMA is a novel, highly comprehensive, supervised, and multi-lingual dataset that supports 7 different tasks in a civil law jurisdiction, privileging generation. Fine-grained task-level comparisons are presented in Section 6.4.B.

6.4.3 Comma

6.4.3.1 Legal Background

CCIR Role and Composition The CCIR, known as "the judge of laws," is responsible for reviewing whether legislative acts have been enacted in accordance with the procedures stipulated in the Constitution and whether their content is consistent with constitutional principles. It is an impartial collegiate body of 15 judges from the Supreme Magistrature. Constitutional judges are elected with a nine-year mandate by the Supreme Tribunals, the Houses of Parliament, and the President of the Republic,

five from each source. They are exclusively committed to the Court work and bring heterogeneous experiences, specializations, and attitudes. A president is elected among the CCIR members (secret ballot, absolute majority) for a 3-year renewable term of office. Importantly, the judges are all equal within the Court. Therefore, decisions are always made collectively and must be considered as the fruit of a collaborative process. The CCIR is not accessible to autonomously choose which questions to examine. Constitutionality questions can be submitted to the CCIR clerk's office through direct appeals from the State/Regions or incidental referrals from any judicial authority.

CCIR Invocation and Decision Making The publication of a legal notice begins the period to present arguments to the Court. A particular CCIR office examines the question and investigates legal precedents. The CCIR President, based on a yearlong calendar, stipulates the docket (case list) for each sitting and appoints a judge (rapporteur) for each case. The rapporteur studies all aspects of the case, prepares a research packet, and prepares a resolution proposal. The CCIR meets in plenary sessions that require the presence of $\geq$11 members, where different judges alternate as rapporteurs. After optional public hearings with the parties' lawyers, the CCIR meets privately to make decisions. Each debate begins with a preliminary statement by the rapporteur and, under the President's direction, continues with the intervention of other judges. Depending on the complexity or controversial nature of the question, the discussion may last a few minutes or entire days (postponing or asking for more material). The President may call for a majority vote to bring the debates to a close. The CCIR can declare three types of decisions: unconstitutional (acceptance), unfounded (rejection), or inadmissible. After the decision, the rapporteur is designated as the author responsible for drafting the Court opinion. Note that the author may have dissented from the majority view. This phase can last anywhere from two weeks to several months. When the draft is ready, the judges collaborate closely to review the document during a chamber conference. If the majority disagrees with the draft, the author may be asked to re-submit it; if the author realizes that there are logical/legal difficulties or uncovered objections, he/she can propose to amend the decision. If none of the judges objects orally or in writing, the order is endorsed by the President and the rapporteur within a few days and filed with the Court's registry, becoming final and public. The CCIR decisions cannot be appealed.

Ruling Types and Structure CCIR rulings, identified by progressive numbers within a year,[23] must be issued as orders or judgments. *Orders* are interlocutory and succinctly motivated instruments designed to solve procedural issues; in CCIR proceedings, they can also close a trial. *Judgments* are fully motivated acts with which the CCIR definitively evaluates. A CCIR ruling consists of three main parts. (i) *Epigraph*, the introductory segment that details panel members, the hearing or the council chamber, the judge rapporteur, and the lawyers heard by the Court. (ii) *Body*, the middle

[23]Since 2016, the CCIR has adopted the European Case Law Identifier (ECLI) in its rulings, allowing for unique identification of all the court judgments at the European level (www.bo-ECLI.eu).

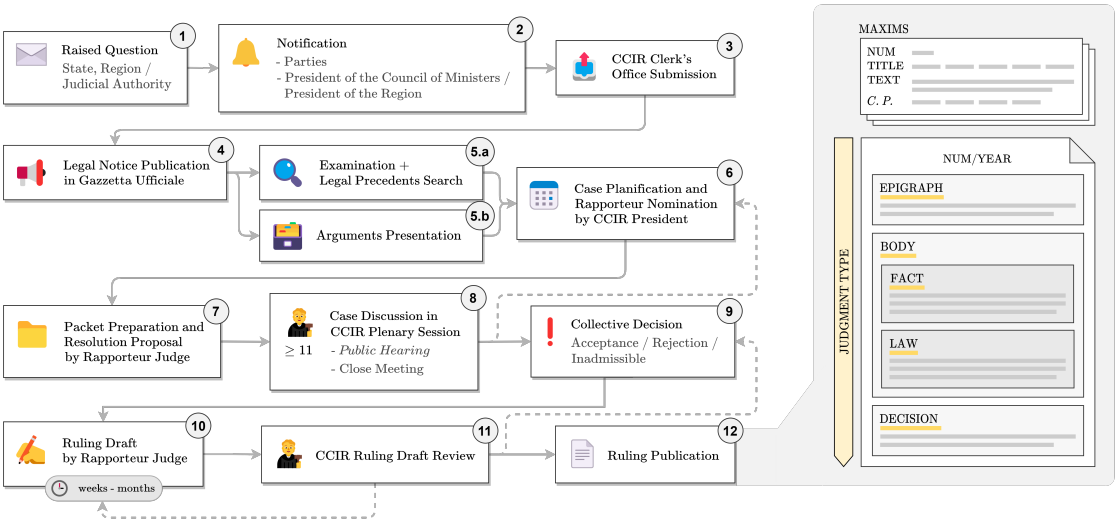


FIGURE 6.25. The writing pipeline for the CCIR rulings contained in COMMA, together with their document structure. Italic text and dashed lines refer to optional procedures and fields, respectively.

section presenting the dispute and the arguments of the parties ("fact"), followed by the legal assessment of the case ("law"). (iii) *Decision*, the concluding part that contains the determination, potentially expressed in many points. Furthermore, a ruling is accompanied by a set of *maxims*, i.e., brief elaborations of the decisions made and their rationale. They comprise an identifier, a title (free-text keyphrase sequence), a text, and an optional set of constitutional parameters—which denote constitutional provisions deemed violated (e.g., "Constitution art. 24 co. 2"). Their n-arity depends on the number of legal issues implied by a case.[24] Lastly, CCIR rulings belong to a *judgment type* (e.g., constitutional legitimacy, suspension of the contested act; details are provided in Appendix 6.4.F). As a result, each order or judgment may extend to a dozen pages. Figure 6.25 sketches the ruling structure and its underlying construction process. We show an example of an order in Section 6.4.G.

CCIR Placement in the International Chessboard The rationale for embracing documents sourced from the CCIR lies in their structural and functional similarity to constitutional courts, tribunals, and supreme courts across diverse legal frameworks worldwide.[25] It is important to note that the CCIR does not operate in isolation; rather, it actively engages with experiences and legal perspectives from other countries. Its primary interactions are within Europe, including countries that share historical and experiential parallels with the Italian Consulta, such as France, Spain, Portugal, and Germany. Collaborative relationships also extend to other regions, particularly Latin

[24] The number of questions decided by the CCIR is usually higher than that of rulings. Not infrequently, a single ruling collects and resolves multiple issues raised by different promotion acts.

[25] https://www.cortecostituzionale.it/jsp/consulta/istituzioni/la_corte_le_altre_corti_EN.do

America. Furthermore, the CCIR is integral to the World Conference on Constitutional Justice. With the growing interaction between Union law and national laws, the European Court of Justice and national courts regularly address analogous legal concerns, and the CCIR is often tasked with resolving international questions. As evidence of this interplay, since 2007, the CCIR has been asserting the unconstitutionality of laws that contradict the European Convention on Human Rights. Therefore, citing Ferrari and Gambero [ctx, 2010], "there are strong reasons to state that in Europe, there is a close interdependence between the levels of judicial review." Recent practice underscores the power of mutual understanding and cooperation between courts. In fact, knowledge sharing strengthens the foundations of constitutionalism, upholding individual rights, achieving balance among state powers, and ensuring judicial protection across boundaries. It is no coincidence that all constitutional case law is indexed on the European Case Law Identifier (ECLI) portal. Facilitating the exploration of legal precedents within the member states of the European Union and in the broader territories is a prominent requirement that actively requires legal translation across many language units.[26] Significantly, since late 2017, the CCIR has ensured access to English-translated versions of a curated set of rulings on the ECLI portal.[27] Therefore, real-world practices already justify the usefulness of translating court decisions into a set of target languages. Although we acknowledge the differences in legal systems, it is crucial to highlight that the significance of Comma transcends the CCIR, contributing to a broader understanding of constitutional justice and its global applications, allowing for cross-lingual exploration. In light of these remarks, we can understand how the models trained on Comma can be easily transferred to constitutional documents from other countries.

6.4.3.2 Tasks

Comma comprises 7 tasks, grouped into 4 families.[28] We list input-output examples in Section 6.4.H. Each task is designed to be meaningful for legal professionals, serving as a valuable measure of text understanding and reasoning.

Abstractive Summarization Full-text CCIR rulings are time consuming to read and digest. Legal case summarization aims to write a short article capturing principal details that are otherwise difficult to come by. The gist of a CCIR ruling can be multi-faced, but—collectively—maxims render a complete picture of the core items. We frame maxims as summaries, helping lawyers in searching arguments for precedence. Comma summaries are in two different granularities:

- **Narrative** (n). Concatenation of maxim texts, each with a paragraph.
- **Bullet Point** (bp). The set of keyphrases in maxim titles, interspersed with ".".

[26] https://curia.europa.eu/jcms/jcms/Jo2_10744/en/

[27] https://dati.cortecostituzionale.it/ECLI/ECLI

[28] As noted by Guha et al. [2022], we emphasize that drawing clear boundaries among Legal NLP tasks can sometimes be difficult, and opinion divergences may exist.

BP targets have received little attention in the NLG literature [Bar-Haim et al., 2020, Shah et al., 2021, Koto et al., 2022, Mahata et al., 2022, Mukherjee et al., 2022]—like generative ruling tags—with no effort in the legal domains. We consider three different summarization tasks. First, we map the rulings (single documents) to each of the summary types above (R → N, R → BP). Then, we attempt to map the maxim texts (concatenated) to their overall titles (N → BP).

Text Generation We assess how well LMs can generate the decision given the ruling epigraph, facts, and the text of the gold constitutional parameters. The network must interpret the provided Constitution articles, apply them to the case background, and predict the outcome of the CCIR. The propositional value of this task lies in helping legal analysis and suggesting dispute resolutions.

Document Classification Ruling categorization is prevalent in the legal profession and requires skills that lawyers are considered to hold distinctly. COMMA supports two single-label classification tasks. The first considers *ruling types*, where the goal is to distinguish orders from judgments. The second stays in a multi-class setting and annotates rulings with 10 *judgment types*. The details of the judgments are given in Section 6.4.F.

Information Retrieval Predicting relevant acts of law, based on a case background, is a crucial research task. Automating this process with semantics-aware techniques can assist lawyers in timely collecting the proper information for verdicts and increase domain familiarity in non-specialist people. Based on the epigraph and fact sections, we evaluate how well LMs can *fetch constitutional parameters* useful for decision-making (with a comma precision level). This task is highly knowledge-intensive because of the necessity of comprehending the provisions' content, other than references. In light of its position at the apex of the regulatory hierarchy, we only consider parameters rooted in Constitution articles, the text of which is broadly accessible.

6.4.3.3 Dataset

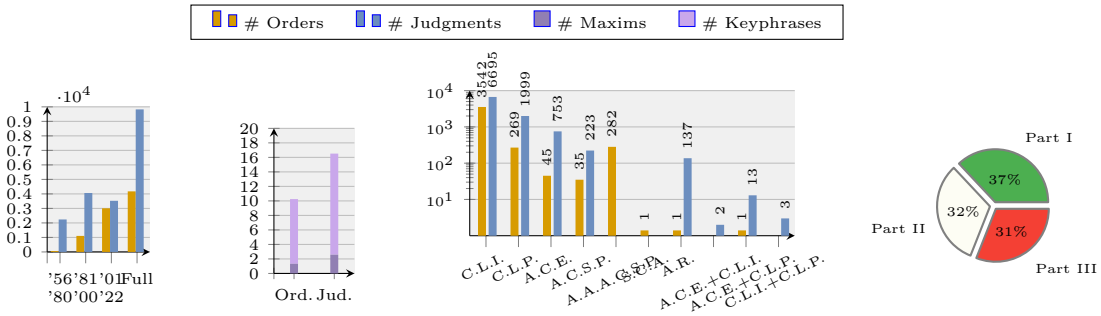
COMMA consists of 14,000 CCIR rulings from 1956 to 2022. Release details are in Appendix 6.4.A. Through extensive preprocessing efforts (Section 6.4.3.4), we compile raw orders, judgments, and maxims—up to April 2022—from the CCIR information repositories[29] and scraped web pages. We split the dataset into training (90%, 12,600), validation (5%, 700), and test (5%, 700), using replacement-free proportional stratified random sampling. To perform information retrieval and judgment prediction tasks, we link COMMA to the full text of the constitutional articles evoked as parameters. Therefore, we cure and publish an electronic version of the Italian Constitution (139

[29]CCIR Open Data Service: https://dati.cortecostituzionale.it/. Data sources are updated weekly and released under the CC BY SA 3.0 license for unrestricted reuse by citizens, researchers, and developers.

TABLE 6.43. COMMA text statistics, all samples. *Left*: task-agnostic average number of words (W) and sentences (S) in rulings, maxims per ruling, and constitutional parameters' text; yielded with NLTK [Bird, 2006]. *Right*: source-target coverage, density, and compression ratio for summarization tasks [Grusky et al., 2018].

	IT		EN		ES		FR	
	W	S	W	S	W	S	W	S
Ruling								
Full-Text	3128.9	99.8	3418.5	97.1	3456.1	87.8	3349.0	110.3
Epigraph	164.8	4.3	184.0	5.4	193.1	4.8	180.2	7.4
Body	2834.4	90.1	3093.6	87.2	3116.8	78.8	3030.4	96.9
Decision	128.8	5.3	141.0	4.6	146.1	4.2	138.3	6.1
Maxims								
Title	116.5	12.4	125.2	12.6	128.9	12.4	123.7	12.4
Text	442.0	18.0	482.5	16.2	487.9	14.4	473.1	17.9
Constitution								
Article	73.5	3.1	77.7	3.1	82.6	3.0	82.3	3.1
Comma	27.5	1.2	29.0	1.2	30.9	1.1	30.8	1.2

	IT	EN	ES	FR
Coverage				
R → N	0.92	0.94	0.92	0.92
R → BP	0.87	0.89	0.87	0.86
N → BP	0.74	0.76	0.74	0.72
Density				
R → N	14.2	11.6	9.9	9.7
R → BP	3.3	3.3	3.0	2.9
N → BP	2.7	2.5	2.2	2.1
Compress				
R → N	8.3	8.3	8.3	8.3
R → BP	31.4	32.3	31.5	31.9
N → BP	4.1	4.2	4.1	4.1



(a) R over the years. (b) Avg maxims & keyphrases per R. (c) R per judgment type (log-scale). (d) Parameters per Constitution part.

FIGURE 6.26. COMMA documents' distribution, all samples. R signifies "rulings."

comma-organized articles divided into 3 parts, 11 titles, and 16 sections).[30] To produce the English, Spanish, and French counterparts of COMMA (i.e., 56,000 documents in total, ∼3 GB), we head-to-head compare various machine translation services and LLMs. After analyzing the back-translation performance (Section 6.4.3.5), we ultimately choose Google Translate. Regarding the Constitution, we extract the official translations provided by the Italian government utilizing advanced scraping techniques. Table 6.43 and Figure 6.26 show our data distribution; details are given in Appendix 6.4.D. Judgments outnumber orders by a ratio of 2.4 to 1, although orders have become more frequent over the years. Each ruling has, on average, 2 maxims and 11 keyphrases across all maxim titles. The types of judgment are imbalanced, with the most popular being cases of constitutional legitimacy and attribution conflict. For classification, we only use those present in all splits (Section 6.4.I.3). In particular, maxims are longer than popular targets in abstractive summarization (e.g., the word count is twice that of BILLSUM [Kornilova and Eidelman, 2019]). Furthermore, maxims and keyphrases are highly source-related

[30] Constitution titles with no subsections count as sections.

(86-94% of their words can be found in extractive fragments from the source rulings), suggesting fewer unsupported entities and facts than datasets with lower coverage. At the same time, the density ranges from 3 to 14, indicating that most summary sentences are not verbatim extractions from the sources. The constitutional parameters equally pertain to all parts of the Constitution, implying that the rulings do not concentrate on specific topics but entail a wide range of questions on fundamental principles (Part I), the Republic organization (Part II), and the rights and duties of citizens (Part III). On average, there are 4 parameters per ruling. The specifications are given in Section 6.4.E. The decision of providing a French and Spanish version of the dataset resides on the fact that Italy, France, and Spain all follow a constitutional framework and are Civil Law countries rooted in Western legal traditions. They have extensive legal codes, including civil law, criminal law, and administrative law. Furthermore, they are influenced by Union law, which includes regulations and directives that member states must implement into their national legal systems. On the other hand, English is the most widely spoken language based on native and non-native speakers. With its recognition as a lingua franca in ongoing court translation initiatives, English inclusion is mandatory. Our decision also came from a linguistic perspective. Italian, French, and Spanish fall under the category of Romance languages. These linguistic affinities are known to facilitate automated translations [Friedland et al., 2022], helping to manage the additional intricacies of legal jargon. Indeed, if two languages share a common ancestor, they are more likely to have similar words for the same semantic concept.

6.4.3.4 Engineering Pipeline

Raw Data CCIR rulings and metadata have been openly released in XML, CSV, and JSON formats (Italian language).[31] In our study, we downloaded the JSON raw archives in April 2022. We merged the files initially organized by years and combined the information on rulings and maxims. This consolidation process yielded 21,429 samples.

Maxim Filtering We reduced the dataset to 21,331 instances by excluding recent rulings that lacked associated maxims.

Size Filtering We retained 14,072 instances that met the criteria of having full texts spanning 1000 to 20,000 words and concatenated maxims ranging from 100 to 2000 words. This filtering process allowed us to exclude texts that deviated significantly in length, ensuring a more balanced dataset.

Duplicate Data Removal Following a similar approach to Kornilova and Eidelman [2019], we applied a deduplication method, resulting in 14,054 instances. Technically, we (i) removed stop words and the 30 most common words (e.g., “article”, “law”, “court”),

[31] https://dati.cortecostituzionale.it/Scarica_i_dati/Scarica_i_dati

(ii) vectorized the texts with `scikit-learn`'s CountVectorizer,[32] (iii) computed the cosine similarity between each pair of rulings (average on full texts and concatenated maxims), (iv) iteratively added rulings to the dataset while discarding examples that were more than 96% similar to any previously included rulings. We found that duplicates predominantly encompassed orders about interconnected topics, released in proximity to one another, or served as amendments to previous rulings with drafting mistakes.

Compression Ratio Filtering Compression measures how much a text is crammed into its summary. Mathematically, this metric represents the ratio between the number of words in the input and its target [Grusky et al., 2018]. Given the wide variation in the sizes of rulings and maxims, we retained instances with a compression ratio between 2 and 70 to avoid the loss of critical information and context. This selection process ended up with 14,000 instances.

Quality Control and Text Cleaning An extensive preprocessing pipeline was implemented to guarantee the textual quality of the dataset. It involved a series of meticulous steps:

- Eliminating unneeded spaces, new line characters, and consecutive punctuation characters.
- Adding missing spaces after punctuation.
- Ensuring proper capitalization.
- Correction of apostrophes in vowels using UTF-8 encoding.
- Removing irrelevant repeated notes at the end of maxims caused by versioning management.
- Excluding the information from the publisher, judge, and reviewer at the end of each decision (not relevant for language modeling).
- Manually correcting typos and structural exceptions.
  - Removing extra slashes and backslashes in article codes because of encoding errors in the original JSON files (e.g., "12//\1983" → "12/1983").
  - Correcting badly formatted dates (e.g., "984" → "1984").
  - Adding missing prefixes (e.g., "for these reasons" keyphrase in decision sections).

Due to their significant role in representing factual information and their frequent occurrence, we made a deliberate decision not to exclude the following elements from the dataset:

- `cf.`: bibliographic citations that refer to external references, providing valuable contextual information.
- `artt.`: legal jargon indicating the citation of multiple articles, contributing to the legal integrity of Comma.
- Names outside the decision signature.

In addition to providing a high-quality full text for each ruling, we secured its structural division into *epigraph*, *body*, and *decision*. Following the same line of work, we further

[32] https://scikit-learn.org

divided maxims in *number*, *title*, and *text*. Some instances (1287 training, 80 validation, 57 test) lack the keyphrases "Considered in Law" and "Believed in Fact," that prevent the division of their body into subsections.

6.4.3.5 Language Translation

Rulings We automatically translate the textual components of the rulings, originally in the Italian language, into English, Spanish, and French, making our work more representative and generalized. In this endeavor, we compare three high-performance machine translation models/services: ChatGPT, Google Translate, and No Language Left Behind (NLLB) [Costa-jussà et al., 2022].

- ChatGPT has already shown great promise as an automatic translator for high-resource European languages [Gao et al., 2023b, Jiao et al., 2023, Peng et al., 2023]. We use the `gpt-3.5-turbo` model and the official API.[33] A new chat session is created for every ruling to ensure that history does not influence translation results. Regarding the decoding hyperparameters, we leave default values (`temperature=1`, `top_p=1`). Complying with the best practices indicated by Weights and Biases (W&B),[34] we use the following prompt messages:

```
[{"role": "system", "content": f"You are a translation assistant. Translate the
    following legal text from {src} to {dest}",
{"role": "user", "content": f"Legal text: {ruling}. Return the translated text only,
     without adding anything else."}]
```

- Google Translate is an internet-based commercial translation product commonly employed in developing NLP multi-lingual benchmarks [Feng et al., 2022a]. We call the Google Translate Ajax API via the `googletrans` library.[35]
- NLLB is a LLM capable of state-of-the-art translations between 200+ languages. Constrained by our hardware (Section 6.4.I.1), we use the 3.3B model in inference mode on a GPU.[36]

We emphasize the dearth of alternatives and the scarcity of competitive translation models supervised for legal text in the languages targeted by our work. The only open-source HuggingFace checkpoints available,[37] are alarmingly small (60M, 512 maximum length) and far from producing outputs of similar quality to the baselines we investigated, albeit not tailored for domain-specific excerpts. We initially allowed for their presence, but subsequently dismissed them because of their evident and intrinsic inferiority, further underlined by the more challenging nature of Comma rulings w.r.t. standard legal training corpora.

Guarding the integrity of standard keyphrases (e.g., "Considered in Law", "Believed in Fact") is paramount for accurately demarcating ruling sections and informing LMs. To

[33] https://platform.openai.com/docs/guides/gpt
[34] https://bit.ly/wandb-doc-translation-gpt4
[35] https://pypi.org/project/googletrans/
[36] https://huggingface.co/facebook/nllb-200-3.3B
[37] SEBIS/legal_t5_small_multitask_it_en, SEBIS/legal_t5_small_multitask_it_fr, SEBIS/legal_t5_small_multitask_it_es

prevent context-dependent translations, we partition the text through Italian section titles and reassemble translated fragments with standardized language-specific keyphrases. Note that for each ruling, we segment the input into multiple parts if it exceeds the maximum length supported by a model or service (e.g., 15K characters for GOOGLE TRANSLATE), allowing us to complete translations in multiple steps. To prevent mid-sentence interruptions, we ensure that each segment concludes at the end of the last complete sentence rather than stopping at the last word. Using two quality control methods, we measure the translation effectiveness across the validation set.

- *Round-Trip Translation* (or Back Translation). It involves translating a text into a different language and re-translating it to the original one. We implement the same approach as Zhu et al. [2019] and Feng et al. [2022a] and measure the ROUGE-1 score [Lin, 2004] by juxtaposing the authentic ruling with the back-translated one.
- *Semantic Similarity*. Considering the legal significance of a maxim, it requires not only overall surface-level quality, but also precise semantic consistency. With this objective in mind, we calculate the sentence-level similarity score between the original maxim and its back-translated counterpart using an Italian cross-encoder.[38]

Table 6.44 presents the results achieved by the models/services in the overall ruling text. Empirically, GOOGLE TRANSLATE is the best option, which also exhibits superior efficiency.[39] Consistent with our results, we draw the attention to the fact that previous research endeavors have effectively harnessed GOOGLE TRANSLATE for multi-lingual content generation (e.g., BLOOMZ, T0) [Muennighoff et al., 2023] and legal translations [WIESMANN, 2019, Galassi et al., 2020, Alkatheery, 2023, Ubayawardena and Hansani, 2023]. Furthermore, we corroborate recent findings that indicate that CHATGPT performance is not on par with commercial translation tools, except when transitioning to GPT-4, which reaches a competitive level [Jiao et al., 2023]. Surprisingly, NLLB did not meet the desired translation quality, generating outputs with a marked low recall. With the hypothesis that the problem could be attributed to the non-moderate length of our documents, we attempted sentence-level translations as an alternative. Still, the scores fell short of expectations. Table 6.45 delves into the effectiveness of GOOGLE TRANSLATE for each language and ruling section, while Figure 6.27 details the ROUGE-1 distributions. Table 6.46 reports a qualitative example of the English translation. Evidently, smaller specialized models are prone to make syntactical errors (e.g., "Prer of the Crtina d'Ampezzo") and distribute subject-predicate-object items into long-range links, thus making text understanding more challenging. The translation of textual descriptions referring to judgment types is performed manually. We keep constitutional parameters unaltered in the normalized form of "`Costituzione art. [art-number] co. [paragraph-number]`."

[38] https://huggingface.co/nickprock/cross-encoder-italian-bert-stsb

[39] It should be noted that we encountered prolonged wait times with the OpenAI API and occasional crashes attributed to request overload.

Table 6.44. Average ROUGE and semantic similarity (SS) % scores on back-translation (full-text rulings and maxims) across all languages. The best results are in bold.

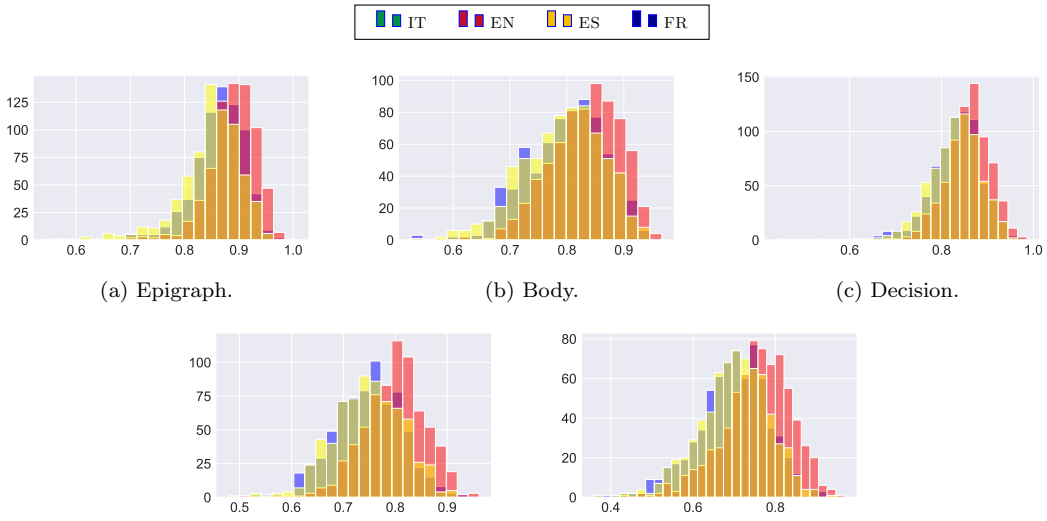
System	ROUGE-1	ROUGE-2	ROUGE-L	SS
ChatGPT	79.75	63.52	76.47	52.20
Google Tr.	**83.69**	**64.05**	**76.88**	37.75
NLLB	60.85	40.51	55.44	**55.07**

Table 6.45. Average Google Translate ROUGE and semantic similarity (SS) scores on back-translation, divided by ruling and maxim section.

	ROUGE-1	ROUGE-2	ROUGE-L	SS
English				
Epigraph	88.99	79.14	88.62	72.68
Body	83.20	71.07	83.95	34.60
Decision	85.90	76.32	87.30	76.56
Maxims	79.77	64.49	80.08	51.35
M. Titles	75.58	57.33	77.12	71.39
Spanish				
Epigraph	84.84	72.35	84.44	71.24
Body	79.25	64.86	79.91	33.56
Decision	83.01	71.64	84.41	75.51
Maxims	74.79	56.83	74.88	49.17
M. Titles	69.90	49.05	71.43	74.05
French				
Epigraph	86.37	74.89	86.12	72.47
Body	79.80	65.32	80.33	33.89
Decision	83.24	72.06	84.67	74.38
Maxims	74.67	56.67	74.78	50.64
M. Titles	69.65	49.02	71.26	72.66

Constitution The Comma translations of the articles of the Italian constitution are derived from manually edited official translations available on reputable platforms such as Wikipedia or digital legal libraries. Precisely, we scrape them from the following web pages: English,[40] Spanish,[41] French.[42] The quality of the translations was evaluated by cross-referencing them with the content found in the PDF files of the Italian Ministry of the Interior, guaranteeing structural consistency and updating the text of the articles as needed.

[40] https://en.wikisource.org/wiki/Constitution_of_Italy
[41] https://es.wikisource.org/wiki/Constituci%C3%B3n_de_la_Rep%C3%BAblica_Italiana
[42] https://mjp.univ-perp.fr/constit/it1947.htm



(a) Epigraph. (b) Body. (c) Decision.

(d) Maxims. (e) Maxims' titles.

FIGURE 6.27. GOOGLE TRANSLATE ROUGE-1 score distributions for ruling sections.

TABLE 6.46. Qualitative example on Italian-to-English machine translation. Decision section, ruling 106/1983.

Original Italian Document	
La Corte costituzionale dichiara non fondata la questione di legittimità costituzionale dell'art. 62, terzo comma, del Testo unico delle leggi di pubblica sicurezza, sollevata dal Pretore di Cortina d'Ampezzo con le ordinanze indicate in epigrafe, in riferimento all'art. 3 della Costituzione. Così deciso in Roma, nella sede della Corte costituzionale, Palazzo della Consulta, il 30 marzo 1983.	
English Translations	
CHATGPT	The Constitutional court declares unfounded the question of constitutional legitimacy of art. 62, third paragraph, of the Consolidated Public Safety Laws, raised by the Magistrate of Cortina d'Ampezzo with the ordinances indicated in the heading, with reference to art. 3 of the Constitution. So decided in Rome, at the headquarters of the Constitutional Court, Palazzo della Consulta, on March 30, 1983.
GOOGLE TRANSLATE	The Constitutional court declares unfounded the question of constitutional legitimacy of the art. 62, third paragraph, of the Consolidated text of public safety laws, raised by the Pretore of Cortina d'Ampezzo with the ordinances indicated in the epigraph, with reference to art. 3 of the Constitution. Decided in Rome, at the seat of the Constitutional Court, Palazzo della Consulta, on 30 March 1983.
NLLB	The Constitutional court finds that the question of the constitutional validity of the third paragraph of Article 62 of the Single European Act on public security, raised by the Pretore of Cortina d'Ampezzo with the orders indicated in the title, in reference to Article 3 of the Constitution, is unfounded.
LEGAL-T5-SMALL	The Constitutional Court declares that the constitutional legitimacy of Article 62, third paragraph, of the text of the single text of public security laws, which was raised by the Prer of the Crtina d'Ampezzo, with the orders indicated in the letter, in relation to Article 3 of the Constitution, and which was decided in Rome, in the seat of the Constitutional Court, the Palace of the Consulta, on 30 March 1983, is not justified by the Constitutional Court.

6.4.4 Experiments

6.4.4.1 Experimental Setup

Models For generative tasks, we offer a baseline using two state-of-the-art multilingual LMs. mBART [Liu et al., 2020, Tang et al., 2021] is an encoder-decoder model largely pre-trained on 50 languages with BART's denoising objective [Lewis et al., 2020a]; it can process inputs up to 1024 tokens in length. Due to the large source content in COMMA, we endow mBART with LSG linear attention [Condevaux and Harispe, 2023], an efficient $\mathcal{O}(n)$ approach that approximates quadratic self-attention with local, sparse, and global attention; it can read up to 4096 tokens. For classification tasks, we employ mBART, using the input representation from the last decoder layer, and XLM-ROBERTA [Conneau et al., 2020], a multi-lingual masked language model pre-trained in 100 languages, empowered with LSG to let it digest up to 4096 tokens. We provide two baseline models for the information retrieval task: XLM-ROBERTA-LSG and a custom solution called MATRIX. The former encodes the input text after replacing every mention of an article of the Constitution with the encoding of its content. The latter performs metric learning with contrastive loss and encodes positive and negative pairs as three-dimensional similarity matrices accounting for the co-occurrence values, the TF-IDF factors, and the embedding representations of the most meaningful words present in the respective instances. Both models return a probability distribution over the search space of articles and commas. Additional technical details are given in Section 6.4.I.

Implementation We fine-tune all models for 4 epochs with PyTorch [Paszke et al., 2019] implementations from the HuggingFace library [Wolf et al., 2019], setting the seed to 42 for reproducibility. In terms of optimization, we use a learning rate of 5e-5 with linear scheduling, AdamW with $\beta_1 = 0.9$ and $\beta_2 = 0.99$, and mixed precision and gradient checkpointing to preserve memory. We select the final checkpoint based on validation performance, operating all samples for text classification and 100 samples for generative tasks to save time. At decoding time, we utilize beam search with 5 beams.

Automatic Evaluation For generative tasks, we consider 7 overall metrics, assessing predictions from a wide spectrum of perspectives. Regarding the syntactic aspect, we embrace ROUGE-{1,2,L} F1 [Lin, 2004] and $\mathcal{R}$ [Moro et al., 2023b] as a combined judgment. BERTScore F1 (BeS) [Zhang et al., 2020b] and BARTScore-$\mathcal{F}$ (BaS) [Yuan et al., 2021] are used to calculate the semantic overlap and the factual consistency, respectively, of the inferred outputs. For classification tasks, we compute μ-F_1 and m-F_1 to consider the problem of class imbalance. For information retrieval, we consider the accuracy (Acc) of the predictions and their ranking scores (NDCG) according to the Italian Constitution section to which they belong. Finally, for every task, we use Carburacy (Υ_m) [Moro et al., 2023b] as a signal to capture both the effectiveness of the model and the eco-sustainability. The technical details are given in Section 6.4.I.4.

Human Evaluation Given the potential failure of automatic metrics and the importance of expert feedback for Legal NLP tasks, we perform an in-depth human evaluation. We provide three legal-expert evaluators with 20 test set rulings and their corresponding gold and machine-generated outputs from mBART-LSG—the best-performed model—for each generative task and language, totaling 320 examples per participant. We randomize the order of instances to guard the rating against being gamed. We indicate the annotators to score predictions p on a Likert scale from 1 (worst) to 3 (best) in the following dimensions: *Recall* considers whether p cover all the target content; *Precision* checks if p covers only target content without including superfluous information; *Faithfulness* examines whether p is factually consistent with the input source. Our setup is exemplified in Section 6.4.K.

6.4.4.2 Results and Discussion

Full Settings A compendium of the results in full settings is available in Table 6.47. The intricate linguistic structure of COMMA is evident in the average disparity between word count and token count across all languages, as seen in the transition from 471 words to 624 tokens in maxim texts. This accentuates the forthcoming requirement for pretraining multi-lingual models—equipped with ad hoc vocabularies—in the legal domain. Due to their ability to accommodate longer inputs, the LSG variant models outweigh the vanilla ones on all four generative tasks. In particular, the advantages of extended input lengths are most evident in tasks involving complete rulings as sources, where reading and comprehending the entire input significantly contribute to improved performance. In R→N, $\mathcal{R}$ ranges from +14% to +28%, BeS from +68% to +96%. In R→BP, $\mathcal{R}$ ranges from +8% to +10%, BeS from +13% to +15%. The performance of the models in N→BP surpasses that in R→BP, further supporting the notion that inputs with more condensed information simplify the summarization task. Unlike French, which registers better BaS values across all tasks, there are no notable shifts in factuality. All models face challenges in producing lengthy summaries (R→N). On average, mBART predictions are around 378 tokens, less than 40% of the ground truths' length of 624. While mBART-LSG learns the intended summary length better, it still stops the prediction at 443 tokens. This draws attention to the shortcomings of current summarization techniques in the delivery of long predictions, which are required by COMMA tasks targeting maxims. According to expectations, an increase in input length and processing time decreases Υ_m, despite the enhanced effectiveness $\mathcal{R}$.

Interestingly, current LMs effectively solve the ruling-type classification task. This can be attributed to the distinct characteristics of the documents, which may demonstrate notable variations in length and language. Unexpectedly, the judgment-type classification is also characterized by high scores. Although to a lesser extent, these findings reaffirm the benefits of linear methods. However, COMMA can be a valuable resource for pretraining [Henderson et al., 2022, Niklaus et al., 2023b] or transfer learning to related tasks in similar legal systems (e.g., English [Fjelstul, 2019], French [Sulea et al., 2017]).

With steady 28% accuracy (excluding encoding-based methods for Spanish), retrieval results signify poor performance. The absence of adequate context is the main factor for

TABLE 6.47. Performance of baseline models (the overall best ones are in bold) on the entire test set of COMMA (700 samples) across all tasks and languages. T denotes the number of generated tokens, which are subwords produced by a subword tokenizer [Kudo and Richardson, 2018].

	R-1	R-2	R-L	$\mathcal{R}$	BeS	BaS	Υ_m	t
AS (r→n)								
IT								
MBART	44.55	23.82	29.10	32.24	20.86	-7.68	53.07	358.9
mBart-Lsg	54.14	34.04	36.53	41.24	33.47	-7.68	48.52	419.5
EN								
MBART	51.15	27.25	33.65	36.97	15.43	-5.37	55.28	388.5
mBart-Lsg	60.55	37.28	40.70	45.70	30.29	-5.93	50.72	412.4
ES								
MBART	52.44	28.29	34.01	37.84	15.89	-7.18	56.09	373.4
mBart-Lsg	58.75	34.81	37.41	43.16	26.75	-7.52	47.48	417.2
FR								
MBART	50.43	26.18	31.10	35.51	16.32	-3.79	54.62	392.2
mBart-Lsg	57.68	33.65	36.00	41.95	28.24	-4.03	61.67	442.5
AS (n→bp)								
IT								
mBart	51.63	31.98	41.25	41.35	40.22	-6.45	61.07	151.5
MBART-LSG	51.96	32.46	41.67	41.77	40.31	-6.43	58.34	148.0
EN								
mBart	54.24	31.88	42.18	42.41	33.96	-5.56	62.28	150.8
MBART-LSG	54.63	32.06	42.33	42.64	34.47	-5.52	60.95	143.0
ES								
mBart	53.83	32.14	41.43	42.13	32.54	-6.50	64.23	145.4
MBART-LSG	54.28	32.50	41.60	42.45	33.19	-6.48	55.74	142.8
FR								
mBart	51.92	30.45	38.78	40.07	32.62	-3.95	63.01	155.2
MBART-LSG	52.68	31.09	39.43	40.74	33.65	-3.97	54.07	157.9

	R-1	R-2	R-L	$\mathcal{R}$	BeS	BaS	Υ_m	t
AS (r→bp)								
IT								
MBART	41.69	23.48	32.90	32.51	30.81	-8.50	51.43	123.6
mBart-Lsg	45.60	27.08	36.19	36.08	34.47	-8.51	46.51	123.6
EN								
MBART	46.49	25.82	36.03	35.86	26.65	-7.33	53.87	124.0
mBart-Lsg	49.67	28.93	38.94	38.90	30.54	-7.32	58.76	128.8
ES								
MBART	46.70	26.08	35.71	35.91	26.35	-8.35	53.76	121.7
mBart-Lsg	50.57	29.94	39.02	39.56	30.06	-8.35	50.59	127.0
FR								
MBART	44.63	24.57	32.96	33.82	26.84	-5.58	52.19	133.0
mBart-Lsg	47.96	27.86	36.03	37.03	30.32	-5.57	57.08	134.5
TG								
IT								
mBart	76.17	68.36	72.43	72.25	65.90	-8.30	78.94	143.6
MBART-LSG	76.46	68.72	72.54	72.50	66.09	-8.30	70.29	143.3
EN								
mBart	77.37	67.82	73.04	72.63	63.86	-6.22	76.27	141.4
MBART-LSG	77.43	68.04	73.21	72.79	64.28	-6.22	72.47	142.2
ES								
mBart	76.21	66.18	71.39	71.14	61.97	-7.57	78.46	148.3
MBART-LSG	76.24	66.29	71.45	71.21	62.04	-7.61	80.44	144.8
FR								
mBart	75.06	65.79	70.41	70.32	62.48	-4.00	77.92	153.5
MBART-LSG	75.14	65.96	70.53	70.44	62.67	-4.04	69.11	153.8

	μ-$\mathbf{F}_1$	m-$\mathbf{F}_1$	Υ_m
SLC (Ruling)			
IT			
MBART	99.14	98.97	90.61
XLM-R-Lsg	100.0	100.0	85.56
EN			
MBART	98.0	97.61	89.74
XLM-R-Lsg	100.0	100.0	88.86
ES			
MBART	99.43	99.32	85.78
XLM-R-Lsg	100.0	100.0	87.72
FR			
mBart	99.57	99.49	90.42
XLM-R-LSG	99.86	99.83	85.65

	μ-$\mathbf{F}_1$	m-$\mathbf{F}_1$	Υ_m
SLC (Judgment)			
IT			
MBART	89.57	35.88	75.74
XLM-R-Lsg	99.57	98.43	88.06
EN			
MBART	98.57	90.58	93.33
XLM-R-Lsg	99.71	99.83	88.20
ES			
mBart	99.71	99.83	95.58
XLM-R-LSG	99.29	95.61	87.29
FR			
mBart	99.43	98.55	95.23
XLM-R-LSG	99.43	98.08	95.41

	Acc	NDCG	Υ_m
IR			
IT			
XLM-R-Lsg	28.50	67.36	84.22
MATRIX	28.50	67.35	84.22
EN			
XLM-R-Lsg	28.50	67.36	84.22
MATRIX	28.50	67.35	84.22
ES			
XLM-R-LSG	34.53	79.73	90.62
MATRIX	28.40	67.51	84.31
FR			
XLM-R-LSG	28.50	67.36	84.22
Matrix	28.53	67.36	84.23

which we claim to be responsible. Even if the text references the parameter, extracting a broad principle like those found in the Constitution from a specific case is challenging. By directing our attention to the ranking, we can observe that the models, although struggling to recognize the specific article, tend to correctly identify the relevant section (semantic domain).

Zooming out, excluding classification, current LMs exhibit nearly adequate performance but still fall short of applicability in real-world settings.

Few-Shot Settings Motivated by the too-costly activity of data labeling in real-world scenarios, we conduct a low-resource study [Moro et al., 2023d,c]. Accordingly, similar to Zhang et al. [2020a], Moro and Ragazzi [2022, 2023], we sample the first k instances

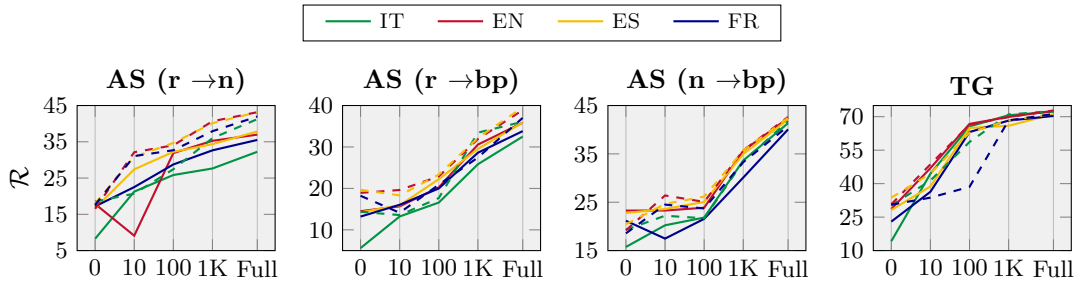


FIGURE 6.28. $\mathcal{R}$ score of MBART (solid) and MBART-LSG (dashed) for each generative task and language with 0, 10, 100, 1000, and full (12,600) training data.

of the training set—with $k \in \{0, 10, 100, 1000\}$—to guarantee reproducibility. In *→BP tasks, the learning curve is much less steep. Even with a few samples, the positive impact of LSG remains evident. Spanish is considered the easiest language to temper. Complete scores are given in Section 6.4.J.

Human Figure 6.29 offers a stack plot with human evaluation results. The average Kendall coefficient among the agreement between the evaluators is 0.18. Although decision generation is ranked highest in terms of automatic metrics, it receives a low factual rating from human evaluators. We recognize a direct proportionality between the input length and the average scores for the Italian language.

6.4.5 Conclusion

We introduce COMMA, a multi-task and multi-lingual Legal NLP dataset collecting 14K rulings from the Constitutional Court of the Italian Republic. Tailored to suit a civil law system, COMMA offers a rich array of features, supporting 7 tasks (4 families), 4 languages, an external supreme law knowledge base, and a curated 66-year archive of long legal documents renowned for their highly-specialized jargon and articulated structure. We perform a rigorous empirical analysis of state-of-the-art language models beyond 10+ automatic metrics and human evaluation, finding substantial room for improvements in abstractive summarization, text generation, and information retrieval.

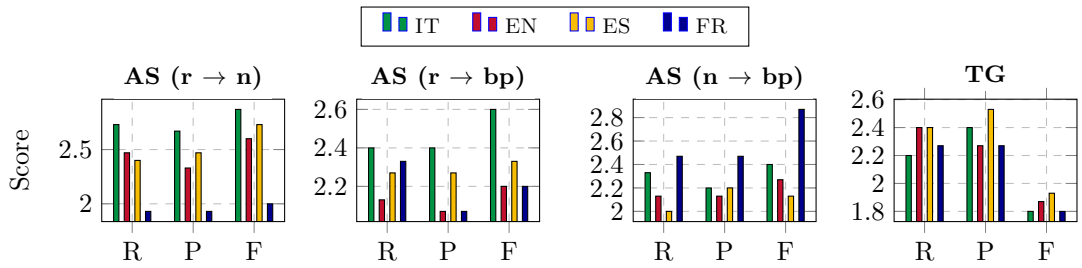


FIGURE 6.29. Human evaluation results on top of MBART-LSG; dimensions: Recall (R), Precision (P), Faithfulness (F).

Limitations and Future Works

COMMA is a wide-ranging project with numerous development opportunities.

- Our external knowledge base consists of Constitution articles only. In the future, it could be extended by integrating additional legal sources. These developments have the capacity to strongly support knowledge-intensive tasks, such as information retrieval, for example, allowing the nearest-neighbor comparison of newly recognized articles with those stemming from the Constitution, or the adoption of retrieval-augmented generation pipelines [Moro et al., 2022a, 2023e]. The importance of this path is counterbalanced by its inherited complexity, which requires greater digital accessibility of legal texts from an open data viewpoint.
- In the pursuit of building more robust models for production purposes, researchers should test various multi-lingual legal pretraining methods, as this area still needs to be scoured. Given its open license, COMMA holds excellent promise for pretraining foundational models in the legal domain.
- Multi-lingual LLMs could be integrated into our model battery, scrutinizing their emerging abilities, reasoning skills, and aptitude for knowledge-intensive tasks.
- To unite international communities and eliminate accessibility barriers, COMMA could be enriched with additional languages.
- Looking ahead, a potential avenue for future exploration involves adopting a summarization pipeline approach to leverage multiple granularities (R $\rightarrow$ N $\rightarrow$ BP), where one model generates a maxim, which is then fed into another model to produce keyphrases. In this paper, we refrain from exploring this avenue due to the established evidence from previous studies [Shen et al., 2022], which has shown that replacing ground truth summaries with ones generated by BART can lead to performance degradation exceeding 20%.
- Future works should look into the temporal dimension.
  - Within COMMA, the articles used as a contextual complement to the rulings consistently pertain to the latest edition of the Italian Constitution (February 2022). Yet, a sounder investigation should use an external time-aware knowledge base associating each ruling with the Constitution articles from the version corresponding to the ruling's date.
  - It would be worthwhile to automatically review how legislative behavior within the CCIR has evolved over time.
  - It is possible to conduct a linguistic analysis to identify the distinctive writing style of the rapporteur judge, which can shed light on the reasons behind the text variations, particularly influenced by the ruling publication year. The results should serve as a foundation for the establishment of guidelines that promote uniformity.

Ethics Statement

We address the nine points explicitly mentioned in the 2023 version of the NeurIPS Ethics Guidelines.

1. Does the data contain any personally identifiable information or sensitive personally identifiable information?

The data may contain PII. Each document follows the privacy rules of the CCIR jurisdiction. This is especially true given that our data are entirely produced by a governmental entity, not merely sanctioned by it. All rulings in COMMA are already public in "Gazzetta Ufficiale", the primary source of information on legislation and official announcements in Italy, and on the institutional CCIR website. However, a comprehensive compilation of these rulings at scale for research purposes is a pioneering endeavor.

When examining the dissemination of judicial pronouncements, it becomes imperative to strike a fair equilibrium between two conflicting interests. On the one hand, there is a need for transparency and publicity of the trial. On the other hand, the privacy protection of the parties involved. The Italian Constitution (art. 111, co. 6) requires that "all judicial decisions must be reasoned," accounting for the decision made to the superior judge, the parties, and the community. This idea dates back to the Neapolitan Enlightenment, specifically to Gaetano Filangieri, who advocated for transcending the *arcana juris*, referring to the veil of mystery that shrouds the law. From this perspective, public oversight is entirely justified, considering that the administration of justice serves not only the interests of the parties, but also the entire national community on behalf of which judgments are rendered. For this control to be effectively exercised, citizens must have the right to know the identities of the parties involved, lawyers, and interests at stake. Only by having this information can they compare the specific ruling with others issued in similar cases and ascertain whether the judge has adhered to the same standard of judgment, irrespective of the parties' significance.

It is no coincidence that, in addition to the cases where anonymization is mandatory, it is essential to have proper justifications supporting any masking done by the CCIR clerk. Therefore, it is appropriate to emphasize that in addition to the cases where the disputed matter can be considered inherently sensitive, it is the responsibility of the applicant to specify the existence and nature of reasons that can justify deviating from the general rule according to which "the dissemination in any form of the content, even integral, of judgments and other judicial provisions is permitted."

For ≈20 years, the CCIR has established a jurisprudence that is very sensitive to privacy protection. In its judicial activity, the CCIR safeguards confidentiality by using initials to mark the parties' names. This marking is done at the beginning and not retrospectively.

In reviewing the CCIR rulings handed down between 2019 and 2020,[43] anonymization emerges as intermittent, possibly influenced by the varying levels of sensitivity among the rapporteur judges (see Section 6.4.3.1). Given the nature of the CCIR's judgments, the issue of confidentiality and the subsequent anonymization of personal data primarily applies to incidental pronouncements and those made in the context of a conflict between powers. Indeed, these are the proceedings in which individuals may be involved (both as

[43] https://www.cortecostituzionale.it/documenti/convegni_seminari/STU%20316_Oscuramento_dati_personali.pdf

parties and as interested third parties). Applying this filter, it is revealed that in 2019, the anonymized rulings represented almost 52%, while in 2020, this figure rose to 58%. When it comes to legal entities, there is a tendency not to anonymize their details unless the name includes elements that enable identification of the individual owner. There are five thematic areas to which anonymization mainly refers.

- *Criminal Procedure.* The scope of these rulings extends to crimes and penalties, trials, the penitentiary system, and surveillance. Individuals involved in these cases, whether defendants, convicts, or prisoners, are subject to mandatory anonymization even before it becomes appropriate. We underline that at times, despite the use of initials in the *epigraph*, the *fact* lacks a specific reference to the person, who is generically referred to as "accused", "sentenced", "detainee", or "applicant."
- *Civil Procedure.* There is a consistent inclination to anonymize the parties' details, especially in cases concerning minors or data related to the status of persons or particular health conditions. However, there are also rulings where the need to protect privacy does not arise as prominently. We refer to judgments concerning the compensation of damages resulting from road traffic, fair compensation, or court costs.
- *Administrative.* Privacy sensitivity is evident in public employment matters, particularly in assigning managerial positions and administrative sanctions. However, unlike criminal cases, a clear and consistent CCIR anonymization approach has not yet been established. Some measures—such as sports sanctions—may not be anonymized despite being sanctions-related. In contrast, there are anonymized provisions on building or construction issues where the need for privacy is not clearly identifiable. In recent judgments concerning the admissibility of conflicts between powers, there is a growing inclination to omit the names of parliamentarians, prioritizing confidentiality over disseminating judicial information.
- *Labor and Social Security.* Most of the issues in this area are related to the recognition of social security benefits in favor of foreigners, minor children, or disability pensions. Even in these cases, the CCIR has frequently opted for greater guarantees, with a growing prevalence of anonymization practices.
- *Tax.* There is no univocal orientation: besides obscured rulings, some present full names.

Court data requests that reproduction exercise due diligence to ensure the accuracy and currency of the materials. As such, we do not redact information further or apply unexpected anonymization filters that might noisily redact factual information or legal citations. Indeed, entirely removing names would break references and comprehension if done non-contextually. For example, a quote such as "The Sicily Region has raised a conflict of attribution against the Bank of Italy *[...]*" (ruling 17/2020) would become "`[MASK]` has raised a conflict of attribution against `[MASK]`," which would not provide any information about the economic sphere nor the regional law to apply. Hence, apart from translated texts, every label in our dataset originates directly from CCIR judges.

2. Does the data contain information that could be deduced about individuals that they have not consented to share?

The case details may include sealed or pseudonymized references that could be used to deduce the identity of the litigants. However, as mentioned above, it is essential to acknowledge that the protection granted to litigants in these cases is in complete accordance with their legal entitlements. Additionally, litigants have ample notice about the degree of protection offered to their identities in public cases and mechanisms to contest anonymity decisions. For instance, if an involved party submits a formal request for their data to be redacted, the clerk of the CCIR, in consultation with the Court President, issues a specific amendment to the Gazzetta Ufficiale. This amendment includes the appropriately anonymized version of the ruling, ensuring its publication while protecting the requested party's information. In any case, we acknowledge that this process may have shortcomings. Thus, following CourtListener [Lissner and Carver, 2010] and Pile of Law [Henderson et al., 2022], we introduce a mechanism to request content removal via HuggingFace Datasets.[44] During this procedure, we kindly request that stakeholders or community submit a takedown or change proposal directly to the upstream data source (i.e., pull request), informing us of the modification and underlying rationales.

3. Does the data encode, contain, or potentially exacerbate bias against people of a certain gender, race, sexuality, or who have other protected characteristics?

The CCIR guarantees equitable treatment and safeguarding of the rights of all individuals, regardless of their gender, race, sexuality, or other protected characteristics. The rigorous selection process for judges and their broad representation aims to prevent conscious or unconscious biases from impacting legal judgments. Consequently, there is a reason to believe that explicit derogatory expressions targeted at vulnerable groups are absent from any ruling. Moreover, our dataset holds significant public interest as it encompasses decisions made by Italy's highest judicial authority and holds binding force for future cases.

4. Does the paper contain human subject experimentation and whether it has been reviewed and approved by a relevant oversight board?

No.

5. Does the paper rely on data that have been discredited by the creators?

No.

6. Consent to use or share the data. Explain whether you have asked the data owner's permission to use or share data and what the outcome was.

[44] https://github.com/huggingface/datasets

The license agreement governing the release of CCIR data (CC BY SA 3.0) implies consent for the use, sharing, and modification, contingent upon proper attribution and the maintenance of the license for subsequent works.

7. Domain-specific considerations when working with high-risk groups

Although high-risk groups are present in this data, we rely on the filtering mechanisms provided by the government publishers of the rulings. While our efforts may enhance the accessibility of data on these groups, it is not apparent that this will result in harm.

8. Filtering of offensive content. For instance, when collecting a dataset, how do authors filter offensive content such as racist language or violent imagery?

Within the dataset, it is possible to encounter instances of toxic or violent language. However, these occurrences arise when the CCIR describes the factual details of a case rather than using explicit derogatory terms. For example, in ruling 141/2019, references are made to "victims of rape" and "recruitment and facilitation of prostitution." This is prevalent in the discussion of criminal cases. We do not filter out such content, as it is vital for legal LMs to comprehend the case's particulars and apply the relevant laws accordingly.

9. Compliance with GDPR and other data-related regulations. For instance, if the authors collect human-derived data, what is the mechanism to guarantee individuals' right to be forgotten (removed from the dataset)?

Our data is sourced from the CCIR, which inherently adheres to the privacy regulations applicable in the EU. However, we offer a content removal mechanism, as explained earlier.

6.4.A Comma Release

Accessing The dataset source files are stored in JSONL format and are publicly available for download through the HuggingFace Datasets platform.[45] For example, to access the English subset of Comma, you need to install the `datasets` Python library and follow the instructions depicted in Figure 6.30. Additional instructions are also available on the project website,[46] which will be regularly updated to incorporate any future changes, updates, or erratum.

License Comma is distributed under the Creative Commons Attribution License (CC BY SA 3.0), while the rulings are already in the public domain. The authors assume all responsibility in the event of a breach of rights.

[45] https://huggingface.co/datasets/[anonymized]/COMMA
[46] https://comma.github.io

TABLE 6.48. Comparison of COMMA to other related public legal text summarization datasets. Measurements include the size of the dataset, the number of words and sentences in the source and target texts, and source-target coverage, density, and compression ratio of words. Except for the number of samples, all reported values are averaged across all instances.

		Source		Target		Source → Target		
Summarization Dataset	**Samples**	W	S	W	S	Cov	Den	Comp
Italian								
EUR-LEX-SUM [Aumiller et al., 2022]	1403	17602.4	421.1	1107.1	32.0	0.85	6.4	16.2
ITA-CASEHOLD [Licari and Comandé, 2022]	1101	4698.2	152.1	800.1	23.2	0.96	113.4	7.2
Comma R → N	14,000	3128.9	99.8	442.0	18.0	0.92	14.2	8.3
Comma R → BP	14,000	3128.9	99.8	116.5	12.4	0.87	3.3	31.4
Comma N → BP	14,000	442.0	18.0	116.5	12.4	0.74	2.7	4.1
English								
BILLSUM [Kornilova and Eidelman, 2019]	23,455	1673.3	46.5	213.0	5.00	0.90	6.9	12.2
GOVREPORT [Huang et al., 2021]	19,463	8765.0	298.7	556.3	18.1	0.94	9.1	17.9
EUR-LEX-SUM [Aumiller et al., 2022]	1504	16390.3	392.0	960.5	30.0	0.87	6.6	17.4
Comma R → N	14,000	3418.5	97.1	482.5	16.2	0.94	11.6	8.3
Comma R → BP	14,000	3418.5	97.1	125.2	12.6	0.89	3.3	32.3
Comma N → BP	14,000	482.5	16.2	125.2	12.6	0.76	2.5	4.2
Spanish								
EUR-LEX-SUM [Aumiller et al., 2022]	1487	17877.3	385.3	1140.4	30.2	0.87	7.1	15.9
Comma R → N	14,000	3456.1	87.8	487.9	14.4	0.92	9.9	8.3
Comma R → BP	14,000	3456.1	87.8	128.9	12.4	0.87	3.0	31.5
Comma N → BP	14,000	487.9	14.4	128.9	12.4	0.74	2.2	4.1
French								
EUR-LEX-SUM [Aumiller et al., 2022]	1505	18076.5	395.4	1190.8	24.1	0.85	6.3	15.4
Comma R → N	14,000	3349.0	110.3	473.1	17.9	0.92	9.7	8.3
Comma R → BP	14,000	3349.0	110.3	123.7	12.4	0.86	2.9	31.9
Comma N → BP	14,000	473.1	17.9	123.7	12.4	0.72	2.1	4.1

Maintenance The authors are committed to providing long-term support for the dataset, monitoring its usage, and developing the necessary bug fixes. For more comprehensive information, we direct readers to the datasheet attached to our website.

6.4.B Task-Level Related Work Comparisons

Table 6.48 and Table 6.49 compare some key statistics between COMMA and the other related text summarization and classification datasets, respectively.

```python
from datasets import load_dataset
# Download COMMA locally and load it as a Dataset
comma = load_dataset("COMMA", "en")
# The first instance of the test set
example = comma["test"][0]
# The full text (epigraph + body + decision)
example["full_text"]
```

FIGURE 6.30. COMMA HuggingFace Dataset.

TABLE 6.49. Comparison of COMMA to other related public legal document classification datasets. Measurements include the size of the dataset, the number of words and sentences in the source texts, and the number of classes and labels. All reported values are averaged across all instances except for the number of samples and classes.

		Source		Target	
Classification Dataset	**Samples**	W	S	C	L
Italian					
MULTI-EURLEX [Chalkidis et al., 2021a]	65,000	1417.1	41.3	21	3.3
Comma SLC (Ruling)	14,000	3128.9	99.8	2	1
Comma SLC (Judgment)	13,980	3125.1	99.7	6	1
English					
LEDGAR [Tuggener et al., 2020]	80,000	127.5	2.1	100	1
MULTI-EURLEX [Chalkidis et al., 2021a]	65,000	1377.1	34.3	21	3.3
Comma SLC (Ruling)	14,000	3418.5	97.1	2	1
Comma SLC (Judgment)	13,980	3414.4	97.0	6	1
Spanish					
MULTI-EURLEX [Chalkidis et al., 2021a]	62,785	1558.8	35.2	21	3.3
Comma SLC (Ruling)	14,000	3456.1	87.8	2	1
Comma SLC (Judgment)	13,980	3451.9	87.6	6	1
French					
MULTI-EURLEX [Chalkidis et al., 2021a]	65,000	1492.5	33.8	21	3.3
Comma SLC (Ruling)	14,000	3349.0	110.3	2	1
Comma SLC (Judgment)	13,980	3345.0	110.2	6	1

6.4.C Constitution Editing/Revising Impact

Since 1947, the Constitution has seen 16 revisions.[47] Our dataset includes rulings from 1956 to 2022. In this timeframe, 5 articles (115, 124, 128, 129, 130) have been abrogated, and 36 articles have undergone modifications. Table 6.50 summarizes all changes and their impact on COMMA.

6.4.D Comma Split Details

We factored in the type and length of the ruling during proportional stratified random sampling. Methodically, this involved (i) ensuring an equal distribution of orders and judgments between the splits to maintain a consistent percentage representation of each type in each split, (ii) dividing them equally based on their lengths, using tertiles to

[47] https://en.wikipedia.org/wiki/Constitution_of_Italy

TABLE 6.50. List of all amendments and their occurrences as constitutional parameters in COMMA rulings; abrogations are highlighted in bold.

Article	Occurrences	Before Amendments	Amendment Year
60	0.01	0	1963
10	1.52	0.04	1967
26	0.01	0	1967
96	0.12	0.03	1989
134	1.87	1.37	1989
135	0.03	0.02	1989
88	0	0	1991
79	0.24	0.09	1992
68	2.53	0.04	1993
48	0.52	0.16	2000
114	1.18	0.07	2001
115	0.29	0.20	2001
116	0.87	0.61	2001
117	20.06	4.01	2001
118	5.33	1.92	2001
119	4.05	0.94	2001
120	2.19	0.14	2001
121	0.47	0.20	2001
122	0.46	0.12	2001
123	0.64	0.18	2001
124	0.03	0.03	2001
125	0.42	0.33	2001
126	0.13	0.07	2001
127	0.86	0.56	2001
128	0.62	0.51	2001
129	0.02	0.01	2001
130	0.25	0.24	2001
131	0.02	0.01	2001
132	0.14	0.03	2001
51	2.01	0.96	2003
81	2.66	1.50	2012
97	13.40	9.97	2012
117	20.06	10.78	2012
119	4.05	2.50	2012
56	0.03	0.03	2020
57	0.04	0.02	2020
59	0.01	0.01	2020
58	0.01	0.01	2021
9	1.10	1.08	2022
41	4.14	4.14	2022

assign them to short, medium, long classes. The high level of balance between splits is evident from the whole distribution statistics presented in Table 6.51 and Figure 6.31.

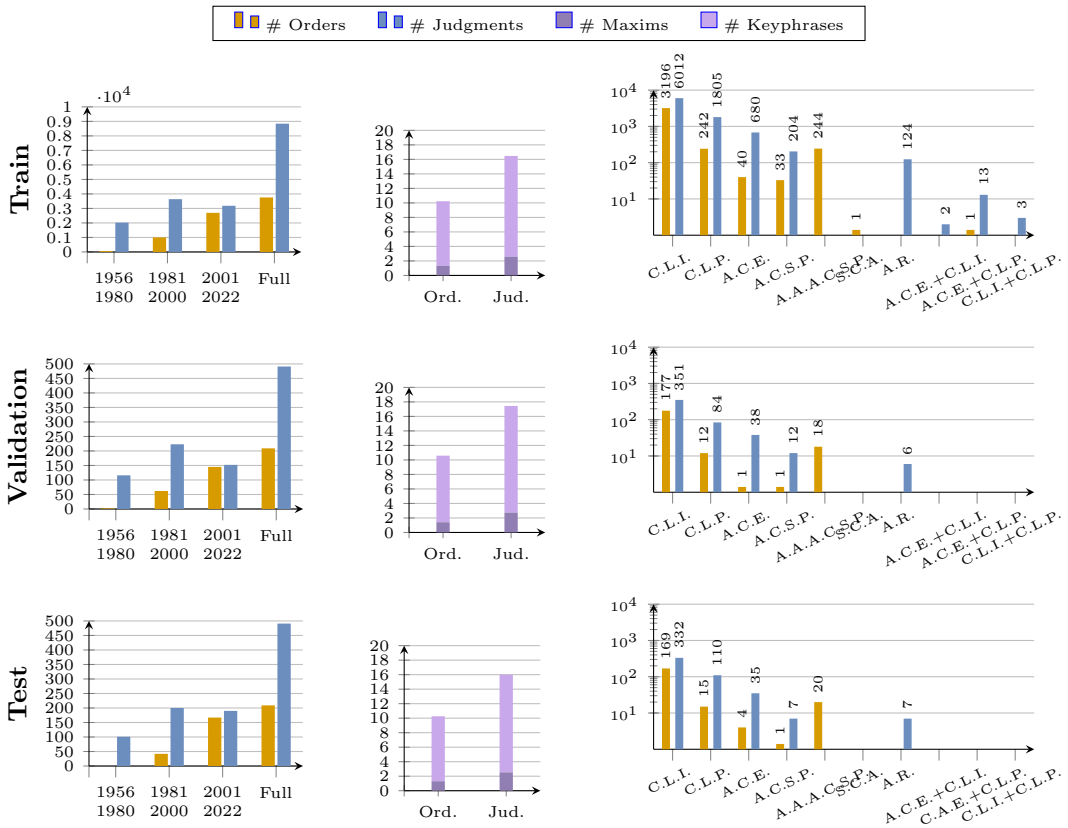
TABLE 6.51. COMMA text statistics across splits. *Left*: task-agnostic average number of words (W) and sentences (S) in rulings, maxims per ruling, and constitutional parameters' text; yielded with NLTK [Bird, 2006]. *Right*: source-target coverage, density, and compression ratio for summarization tasks [Grusky et al., 2018].

	IT		EN		ES		FR	
	W	S	W	S	W	S	W	S
Train								
Ruling								
Full-Text	3125.2	99.7	3414.7	97.1	3452.2	87.7	3345.1	110.2
Epigraph	164.4	4.3	183.6	5.4	192.7	4.8	179.8	7.4
Body	2831.9	90.0	3091.0	87.1	3114.2	78.8	3027.8	96.8
Decision	128.1	5.2	140.2	4.6	145.4	4.2	137.6	6.0
Maxims								
Title	116.3	12.4	125.0	12.6	128.7	12.4	123.6	12.4
Text	440.6	17.9	481.1	16.1	485.9	14.3	471.6	17.9
Validation								
Ruling								
Full-Text	3190.6	100.6	3482.2	97.3	3519.2	88.1	3411.1	110.8
Epigraph	172.2	4.4	191.2	5.3	200.1	4.7	187.4	7.4
Body	2879.7	90.4	3140.5	87.2	3163.3	79.1	3075.8	97.1
Decision	138.0	5.7	150.6	4.9	155.9	4.4	147.9	6.4
Maxims								
Title	122.2	13.1	131.2	13.3	134.9	13.0	129.6	13.0
Text	469.4	19.4	512.4	17.3	518.0	15.3	502.2	18.9
Test								
Ruling								
Full-Text	3132.8	100.9	3423.3	98.0	3462.1	88.0	3356.2	111.5
Epigraph	164.8	4.4	184.8	5.5	194.1	4.9	181.2	7.6
Body	2835.0	91.0	3094.0	87.9	3117.6	78.9	3033.0	97.8
Decision	132.0	5.3	144.5	4.7	150.4	4.3	142.0	6.2
Maxims								
Title	113.6	12.1	121.7	12.3	125.4	12.1	120.4	12.1
Text	439.4	18.4	478.9	16.4	483.8	14.3	470.4	18.1

	IT	EN	ES	FR
Train				
Coverage				
R → N	0.92	0.94	0.92	0.92
R → BP	0.87	0.89	0.87	0.86
N → BP	0.74	0.76	0.74	0.72
Density				
R → N	14.2	11.6	9.9	9.7
R → BP	3.3	3.3	3.0	2.9
N → BP	2.7	2.5	2.2	2.1
Compress				
R → N	8.3	8.3	8.3	8.3
R → BP	31.5	32.4	31.6	31.9
N → BP	4.1	4.2	4.1	4.1
Validation				
Coverage				
R → N	0.92	0.94	0.92	0.92
R → BP	0.87	0.89	0.87	0.86
N → BP	0.74	0.77	0.74	0.72
Density				
R → N	13.4	10.9	9.3	9.3
R → BP	3.3	3.3	3.0	2.9
N → BP	2.7	2.6	2.3	2.1
Compress				
R → N	7.9	8.0	8.0	8.0
R → BP	29.7	30.5	29.8	30.1
N → BP	4.1	4.2	4.1	4.1
Test				
Coverage				
R → N	0.92	0.94	0.92	0.92
R → BP	0.87	0.89	0.87	0.86
N → BP	0.74	0.76	0.74	0.72
Density				
R → N	14.3	11.7	10.0	9.9
R → BP	3.4	3.4	3.1	3.0
N → BP	2.7	2.6	2.3	2.2
Compress				
R → N	8.3	8.4	8.3	8.3
R → BP	32.0	32.9	32.1	32.5
N → BP	4.1	4.2	4.1	4.2

6.4.E Insights on the Distribution of Constitutional Parameters

Figure 6.32 and Figure 6.33 complement the statistics per part outlined in the main paper (Section 6.4.3.3). They provide a deeper examination of the apportionment of the Constitution titles and sections, shedding light on the prevailing Constitution articles utilized as parameters by the CCIR. A considerable portion of constitutional parameters relates to the core principles of the Italian legal system, which cover the inviolable rights of individuals, social rights, equality, protection of linguistic minorities, family matters, labor regulations, and environmental preservation. The three most popular articles are 3 (equality principles among individuals), 24 (freedom of expression), and 117 (division of powers between the central government and regional authorities).

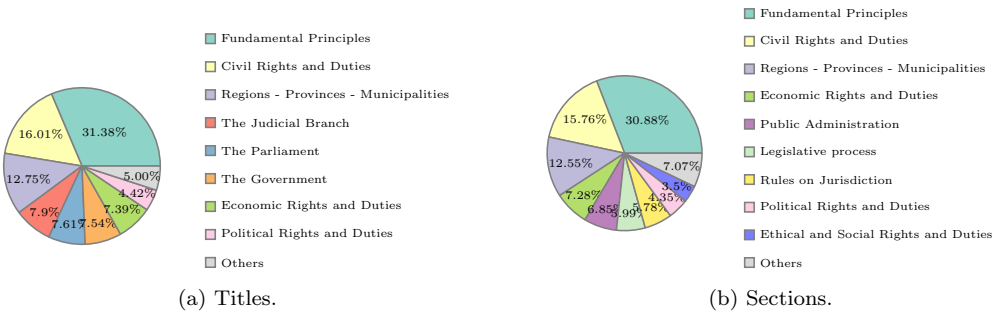


(a) Rulings over the years.

(b) Avg maxims and keyphrases per ruling.

(c) Rulings per judgment type (log-scale).

FIGURE 6.31. COMMA DOCUMENTS' DISTRIBUTION ACROSS SPLITS.



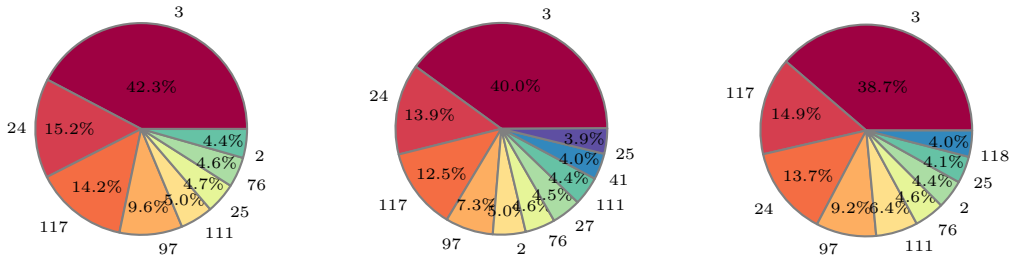
(a) Titles.

(b) Sections.

FIGURE 6.32. DISTRIBUTION OF CONSTITUTIONAL PARAMETERS AMONG CONSTITUTION TITLES AND SECTIONS ACROSS ALL SPLITS.

6.4.F CCIR Judgment Types

Table 6.52 summarizes the 10 types of judgment used by the CCIR and their corresponding definition.



(a) Training. Frequency > 700. (b) Validation. Frequency > 40. (c) Test. Frequency > 40.

FIGURE 6.33. Distribution of Constitution articles that occur most frequently as constitutional parameters in COMMA rulings. Statistics are divided by split.

6.4.G CCIR Ruling Example

Figure 6.34 depicts a complete COMMA instance. More examples can be inspected using the preview functionality offered on the online HuggingFace dataset card.

6.4.H Input-Output Task Examples

Figure 6.35 displays training examples, i.e., input(s)-output pairs for each task in COMMA.

6.4.I Implementation Details

6.4.I.1 Computational Resources and Environment

Each experiment is performed on an internal workstation using a single Nvidia GeForce RTX3090 GPU with 24 GB of dedicated memory, 64 GB of RAM, and an Intel® Core™ i9-10900X1080 CPU @ 3.70GHz. The reference operating system is Ubuntu 20.04.3 LTS. Our development environment is constructed on top of a docker container with a de-facto standard HuggingFace image to elevate portability and consistency.[48] We use HuggingFace v.4.30.0.

6.4.I.2 Models

Table 6.53 enumerates the model checkpoints used in this study. As regular in generative tasks, we define a maximum output length to avoid the production of a EOS token after a certain threshold. We determine the 512 max-length constraints by examining the output sizes in the validation set. We do not restrict the minimum number of tokens to generate, allowing models to learn it in a training time, a challenging task for long legal targets [Shen et al., 2022]. When conducting zero-shot experiments, the generated text lengths mainly rely on the pretraining settings, since the models have not undergone training on the downstream tasks. To ensure fair comparisons and improved control over the length of the output, following Zhu et al. [2021] and Xiao et al. [2022], we establish

[48] https://hub.docker.com/r/huggingface/transformers-pytorch-gpu

TABLE 6.52. Judgment types for COMMA rulings.

#	Name	Acronym	Description
1	Judgment of Constitutional Legitimacy in an Incidental Way	C.L.I.	Judgment of constitutionality promoted, during a trial, by a judge who asks the Court to examine the constitutional legitimacy of a law to which it must apply.
2	Judgment of Constitutional Legitimacy in a Principal Way	C.L.P.	Judgment of constitutionality promoted by the State against a regional law or by a Region against a law of the State or another Region.
3	Judgment for Attribution Conflict between Entities	A.C.E.	Judgment promoted by the State (or by a Region) so that the Court protects its constitutionally-guaranteed faculties against invasive (non-legislative) acts implemented by a Region (or by the State or by another Region).
4	Judgment for Attribution Conflict between State Powers	A.C.S.P.	Judgment promoted by a State power so that the Court protects its constitutionally-guaranteed faculties against the invasive acts, usually non-legislative, of another power. The procedure is divided into two phases. In the first, the Court adopts an order in chambers that evaluates the admissibility of the conflict proposed by the appellant Authority. In the second, concerning the merits of the dispute, the Court establishes with a sentence which power has the attribution in dispute.
5	Judgment on the Appeal Admissibility for Attribution Conflict between State Powers	A.A.A.C.S.P.	Judgment with which the Court declares admissible or inadmissible a conflict question about the powers and competencies of different branches of government. The Court aims to ensure the balance and separation of powers in the State.
6	Judgment for the Suspension of the Contested Act	S.C.A.	Judgment on the suspension of a legislative or regulatory act pending a final decision on its constitutionality. It aims to prevent the contested act from causing harmful or irreversible effects before its conformity with the Constitution is determined.
7	Judgment on the Admissibility of Referendums	A.R.	Judgment with which the Court declares admissible or inadmissible a referendum question, already deemed regular by the Central Office of the Court of Cassation, aimed at repealing a state law in whole or in part.
8	Judgment for Attribution Conflict between Entities + Judgment of Constitutional Legitimacy in an Incidental Way	A.C.E.+C.L.I.	See (3) and (1).
9	Judgment for Attribution Conflict between Entities + Judgment of Constitutional Legitimacy in a Principal Way	A.C.E.+C.L.P.	See (3) and (2).
10	Judgment of Constitutional Legitimacy in an Incidental Way + Judgment of Constitutional Legitimacy in a Principal Way	C.L.I.+C.L.P.	See (1) and (2).

a maximum length limit at inference time, which is set to the average length of the gold targets (rounded to the nearest power of 2). The only exception to this rule is the R→N abstractive summarization task, which expects longer predictions.

In this paper, we do not explore extractive summarization models due to the abstractive nature of the targets in COMMA (Table 6.43) and the findings of recent studies [Shen et al., 2022], where extractive approaches perform poorly despite targets with high coverage in the source (92-96%).

TABLE 6.53. List of the utilized models and relative length constraints.

Model	URL	#Params	Input	Output	
			Max Len	Min Len	Max Len
MBART	facebook/mbart-large-50	∼611M	1024	0	512 (128 for zero-shot tasks except R→N)
MBART-LSG	facebook/mbart-large-50 + LSG	∼611M	4096	0	512 (128 for zero-shot tasks except R→N)
XLM-R-LSG	xlm-roberta-base + LSG	∼281M	4096	/	/

6.4.I.3 Task Details

Decision Generation When ruling body sentences cannot be categorized as "Believed in Fact" or "Considered in Law" (see Section 6.4.3.4), we deliberately omit those instances from the training set to avoid providing a legal application context that could potentially assist in predicting the CCIR decision.

Judgment Type Classification As anticipated in the main text (Section 6.4.3.3), we classify only judgment types with at least one instance in each split. Discarded classes are S.C.A., A.C.E.+C.I.I., A.C.E.+I.C.P., C.I.I.+C.I.P. The density scores in Table 6.54 point to the absence of explicit references to the types of judgment within the full text of the rulings. These results defend the score in Table 6.47, unveiling the lack of conspicuous patterns that could facilitate the prediction of LMs.

Information Retrieval We have approached the information retrieval task in two different ways. The first, labeled XLM-R-LSG in Table 6.47, uses a baseline classification model to encode the input instance, which includes the Epigraph and "Believed in Fact" sections. It then transforms this encoded representation into a cumulative distribution over the search space, composed of articles and commas extracted from the Italian Constitution. Instead of relying on traditional keyword-based search or embedding similarity methods, our proposal leverages the power of classification algorithms to assign a probability measure to each article and a comma based on their relevance to the input instance. This allows us to rank the output distribution according to the article subsets of interest. To provide the text with additional contextual and semantic input cues that the encoder can use to base its predictions, we substitute every mention of articles from the Constitution with the encoded representation of their content. By doing

TABLE 6.54. Character-level ruling-judgment type density compared to the total judgment type length. Italian language.

Judgment	Chars	Density
C.L.I.	57	2.34
A.C.E.	47	0.96
A.A.A.C.S.P.	91	1.06
C.L.P.	56	2.40
A.C.S.P.	61	1.27
A.R.	42	0.51

so, we aim to streamline the information retrieval process and facilitate access to precise and pertinent information, which is paramount in technical and formal documents like those presented here. The second architecture, named MATRIX in Table 6.47, learns the representation of text by metric learning with the contrastive loss $\mathcal{L}_{contr}$ defined as:

$$\text{dist}_{x_1,x_2} = \|W_\theta x_1 - W_\theta x_2\|_2^2$$

$$\mathcal{L}_{contr} = \begin{cases} dist_{x_1,x_2} & \text{if } y_1 = y_2 \\ \max(0, \alpha - dist_{x_1,x_2}) & \text{if } y_1 \neq y_2 \end{cases}$$

where α (set to 1.0 for all our experiments) is the margin defining the minimum desired distance in Euclidean terms of the positive and negative representations. Similarly to the first approach, this model outputs a distribution over the search space. In this context, positive and negative samples are defined on the basis of the Constitution's section of the articles reported in the constitutional parameters. This scheme bases its analysis on the distribution of words in the input texts. To assess the representativeness of the data, we used the TF-IDF index to construct a vocabulary of significant words derived from the ruling corpus, excluding stop words and uninformative legal phrases.

Drawing on the insights presented in Rosso-Mateus et al. [2020], which exploits the particularity of text from a syntactical point of view to retrieve biomedical passages, we convert each anchor-positive and anchor-negative pair into three-channel similarity matrices of size $n \times n$. To accomplish this, we compute the cross-product among the occurrence count, TF-IDF value, and embedded representation. The embedded representation is obtained using the embedding layer of the XLM-ROBERTA-LSG model. This computation is performed for the top-n significant words identified in the input text of each instance in the pair. This process results in a three-dimensional encoding for each sample. Unlike the first approach, which is based on the encoding of textual data, this model employs a Convolutional Neural Network to process the similarity matrices and derive the representation of the "Epigraph" and "Believed in Fact" inputs. To extract relevant information from the input text and create helpful feature representations for the metric learning task, we deploy a Siamese Network that, by sharing the weights for the instances in pairs, learns to generate meaningful and discriminative feature representations regardless of whether the sample is positive or negative. This last approach is designed as an end-to-end network that simultaneously learns the representation of input data and the final retrieval prediction using the weighted loss function $\mathcal{L}_{e2e}$ that combines contrastive loss $\mathcal{L}_{contr}$ and cross-entropy loss $\mathcal{L}_{ce}$.

$$\mathcal{L}_{e2e} = \lambda \cdot \mathcal{L}_{ce} + (1 - \lambda) \cdot \mathcal{L}_{contr}$$

with the hyperparameter λ set to 0.6 for all our experiments. Such a holistic learning approach enables the model to optimize its performance and derive meaningful representations while delivering consistent results in the retrieval task.

Regarding the input data, note that the presence of numerical discrepancies in articles does not invalidate the entirety of an instance, as it typically involves only one of the multiple articles mentioned in the constitutional parameters. We appropriately

addressed and rectified such erroneous information during the data preprocessing phase for this task.

6.4.I.4 Metrics

We quantify automatic metric scores using NLG-Metricverse [Frisoni et al., 2022a]; if not possible, we resort to external official repositories. Table 6.55 lists all hyperparameters.

6.4.I.5 Tracking

We track all our runs with Weights & Biases (`https://wandb.ai`) and monitor CO_2 emissions with CodeCarbon (`https://github.com/mlco2/codecarbon`). Details are given in Section 6.4.L.

6.4.J Low-Resource Details

Table 6.56 and Table 6.57 show the metric scores for each combination of generative tasks, language, model, and sample size. In zero-shot scenarios, Lsg generates extremely concise text, whereas mBart often reaches the max length due to its pretraining.

6.4.K Human Evaluation Details

Table 6.58 clarifies the interpretation of each Likert scale score related to quality criteria. By minimizing annotation ambiguity and subjectivity, this approach secures the attainment of comparable results and facilitates a reproducible and objective evaluation. Our interface is shown in Figure 6.36, inspired by Gu et al. [2022].

6.4.L Carbon Impact Statement

As suggested by Lacoste et al. [2019] and Henderson et al. [2020], we report the carbon impacts of our experiments in Table 6.59.

ORDER 203/2011

JUDGMENT OF CONSTITUTIONAL LEGITIMACY IN AN INCIDENTAL WAY

MAXIM	
NUMBER	35721
TITLE	• taxes • tax breaks • tariff facilitation for the purchase of the "first home" • conditions for application. declaration by the purchaser that he is not the owner of property, usufruct, use and habitation rights over another house in the territory of the same municipality in which the purchased property is located • alleged unreasonable discrimination against buyers who hold identical rights to residential properties located in different municipalities, with violation of the principles of rationality and non-contradiction • absolute lack of motivation on the non-manifest groundlessness and insufficient motivation on the relevance • manifest inadmissibility of the question
TEXT	**The question of constitutional legitimacy of letter b) of number 1) of note II-bis of art. 1 of Part One of the Tariff attached to Presidential Decree 26 April 1986, no. 131**-as referred to in the first period of number 21) of Part II of Table A attached to the Presidential Decree 26 October 1972, no. 633-**is manifestly inadmissible given**, in relation to the aforementioned art. 53 of the Constitution, **the absolute lack of motivation on the non-manifest groundlessness, and**, with reference to art. 3 of the Constitution and the principles of reasonableness, rationality and non-contradiction, **the incomplete description of the case object of the main proceedings with consequent insufficient motivation on the relevance.**
C.P.	Constitution art. 3; Constitution art. 53

RULING	
EPIGRAPH	In the judgment of constitutional legitimacy of letter b) of number 1) of note II-bis of art. 1 of Part One of the Tariff attached to Presidential Decree 26 April 1986, no. 131 (Approval of the consolidated text of the provisions concerning the registration tax) *[...]* promoted by order filed on 16 February 2010 by the Provincial Tax Commission of Milan in the proceedings pending between Maria Ricco Galluzzo and the Revenue Agency, office of Milan 6 *[...]* published in the Official Gazette of the Republic n. 45, the first special series of the year 2010. Given the act of intervention of the President of the Council of Ministers; having heard Judge Rapporteur Franco Gallo in chambers on 11 May 2011."
BODY	**BELIEVED IN FACT** **Considering that**, with an order pronounced on November 20, 2009 and filed on February 16, 2010, the Provincial Tax Commission of Milan *[...]* raised, with reference to articles 3 and 53 of the Constitution and the principles of reasonableness, rationality, and non-contradiction, the question of the legitimacy of letter b) of number 1) of note II-bis of art. 1 of Part One of the Tariff attached to Presidential Decree 26 April 1986, no. 131 (Approval of the consolidated text of the provisions concerning the registration tax); **that** *[...]* on the basis of the contested provision, for the purposes of applying the so-called tax concessions for the first home "to transfer deeds for consideration of the property of non-luxury residential houses and to deeds transferring or constituting bare ownership, usufruct, of the use and residence relating to the same, the following conditions must apply: *[...];* **that**, according to what was reported by the referring judge: a) *[...]* b) *[...]*, c) *[...]*, d) *[...];* **that** the same referring court argues that individuals who hold property rights, usufruct, use, and habitation of a house, without benefitting from the aforementioned tax concessions, are subject to unfair treatment when they purchase another home and relocate depending on whether the newly purchased home is in the same municipality or a different one; **that** in fact - continues the referring judge - the purchase of the first home is considered as a circumstance impeding the application of the concessions for the purchase of the second home only in the case in which both homes are located in the same municipal area *[...];* *[...]* **CONSIDERED IN LAW** **Considering that** the Provincial Tax Commission of Milan doubts - with reference to articles 3 and 53 of the Constitution and the principles of reasonableness, rationality and non-contradiction - the legitimacy of letter b) of number 1) of note II-bis of art. 1 of Part One of the Tariff attached to Presidential Decree 26 April 1986, no. 131 *[...]*; **that** *[...]* the purchaser declares that he is not the exclusive owner or in communion with the spouse of the property rights, usufruct, use, and habitation of another house in the territory of the municipality in which the property to be purchased is located"; *[...]* **that** the question - as objected by the Attorney General of the State - is manifestly inadmissible; **that**, as regards the aforementioned art. 53 of the Constitution, the inadmissibility derives from the absolute lack of motivation on the non-manifest groundlessness because the referring court limited himself to indicating the aforementioned article of the Constitution; **that**, as for the other parameters, the inadmissibility derives from the incomplete description of the case subject to the main proceedings and, therefore, from the insufficient reasoning on materiality; ***[...]*** **that** the referring court: a) does not clarify whether the house owned by the taxpayer since 2003 is concretely suitable for satisfying the housing needs of the interested party; b) does not consider that *[...]* the contested provision *[...]* must be understood in the sense that the ownership of a residential house constitutes an obstacle to the use of tax breaks for the subsequent purchase of another house located in the same Municipality only if the first of the two houses is already suitable to satisfy the housing needs of the interested party *[...]*; c) does not specify whether the declaration made by the taxpayer in 2006, at the time of the purchase of the second home, that she does not own another house in the same municipal area *[...]* must be considered false *[...];* **that** the found inadmissibility of the question prevents the detection of its manifest groundlessness, deriving from the fact that the concessions in question respond to the reasonable ratio of favoring the purchase of a home in the place of residence or work for the benefit of those who, in the same place, do not own another house.
DECISION	**The Constitutional Court declares the manifest inadmissibility of the question of constitutional legitimacy** of letter b) of number 1) of note II-bis of art. 1 of Part One of the Tariff attached to Presidential Decree 26 April 1986, no. 131 (Approval of the consolidated text of the provisions concerning the registration tax), as referred to by the first sentence of number 21) of Part II of Table A attached to the Presidential Decree 26 October 1972, no. 633 (Institution and regulation of the value added tax), raised, with reference to the articles 3 and 53 of the Constitution and the principles of reasonableness, rationality and non-contradiction, by the Provincial Tax Commission of Milan with the ordinance indicated in the epigraph. Thus decided in Rome, in the seat of the Constitutional Court, Palazzo della Consulta, on 22 June 2011.

FIGURE 6.34. English ruling example from the COMMA training set, order 203/2011.

<table>
<tr><th colspan="3">ABSTRACTIVE SUMMARIZATION</th></tr>
<tr><td>R→N
 
</td><td>In the judgment of constitutional legitimacy of letter b) of number 1) of note II-bis of art. 1 of Part One of the Tariff attached to Presidential Decree 26 April 1986, no. 131 (Approval of the consolidated text of the provisions concerning the registration tax) [...] BELIEVED IN FACT. [...] CONSIDERED IN LAW. [...] The Constitutional Court declares the manifest inadmissibility of the question [...] raised, with reference to the articles 3 and 53 of the Constitution and the principles of reasonableness, rationality, and non-contradiction, by the Provincial Tax Commission of Milan with the ordinance indicated in the epigraph. Thus decided in Rome, in the seat of the Constitutional Court, Palazzo della Consulta, on 22 June 2011.</td><td>The question of constitutional legitimacy of letter b) of number 1) of note II-bis of art. 1 of Part One of the Tariff attached to Presidential Decree 26 April 1986, no. 131-as referred to in the first period of number 21) of Part II of Table A attached to the Presidential Decree 26 October 1972, no. 633-is manifestly inadmissible given, in relation to the aforementioned art. 53 of the Constitution, the absolute lack of motivation on the non-manifest groundlessness, and, with reference to art. 3 of the Constitution and the principles of reasonableness, rationality and non-contradiction, the incomplete description of the case object of the main proceedings with consequent insufficient motivation on the relevance.</td></tr>
<tr><td>R→BP
 
</td><td>In the judgment of constitutional legitimacy of letter b) of number 1) of note II-bis of art. 1 of Part One of the Tariff attached to Presidential Decree 26 April 1986, no. 131 (Approval of the consolidated text of the provisions concerning the registration tax) [...] BELIEVED IN FACT. [...] CONSIDERED IN LAW. [...] The Constitutional Court declares the manifest inadmissibility of the question [...] raised, with reference to the articles 3 and 53 of the Constitution and the principles of reasonableness, rationality, and non-contradiction, by the Provincial Tax Commission of Milan with the ordinance indicated in the epigraph. Thus decided in Rome, in the seat of the Constitutional Court, Palazzo della Consulta, on 22 June 2011.</td><td>• taxes
• tax breaks
• tariff facilitation for the purchase of the "first home"
• [...]
• absolute lack of motivation on the non-manifest groundlessness and insufficient motivation on the relevance
• manifest inadmissibility of the question</td></tr>
<tr><td>N→BP
  </td><td>The question of constitutional legitimacy of letter b) of number 1) of note II-bis of art. 1 of Part One of the Tariff attached to Presidential Decree 26 April 1986, no. 131-as referred to in the first period of number 21) of Part II of Table A attached to the Presidential Decree 26 October 1972, no. 633-is manifestly inadmissible given, in relation to the aforementioned art. 53 of the Constitution, the absolute lack of motivation on the non-manifest groundlessness, and, with reference to art. 3 of the Constitution and the principles of reasonableness, rationality and non-contradiction, the incomplete description of the case object of the main proceedings with consequent insufficient motivation on the relevance.</td><td>• taxes
• tax breaks
• tariff facilitation for the purchase of the "first home"
• [...]
• absolute lack of motivation on the non-manifest groundlessness and insufficient motivation on the relevance
• manifest inadmissibility of the question</td></tr>
<tr><th colspan="3">TEXT GENERATION</th></tr>
<tr><td>DECISION GENERATION
 
</td><td>In the judgment of constitutional legitimacy of letter b) of number 1) of note II-bis of art. 1 of Part One of the Tariff attached to Presidential Decree 26 April 1986, no. 131 (Approval of the consolidated text of the provisions concerning the registration tax) [...] BELIEVED IN FACT. [...]
[Constitution Article 3 Text]
[Constitution Article 53 Text]</td><td>The Constitutional Court declares the manifest inadmissibility of the question [...] raised, with reference to the articles 3 and 53 of the Constitution and the principles of reasonableness, rationality, and non-contradiction, by the Provincial Tax Commission of Milan with the ordinance indicated in the epigraph. Thus decided in Rome, in the seat of the Constitutional Court, Palazzo della Consulta, on 22 June 2011.</td></tr>
<tr><th colspan="3">DOCUMENT CLASSIFICATION</th></tr>
<tr><td>RULING TYPE
 
</td><td>In the judgment of constitutional legitimacy of letter b) of number 1) of note II-bis of art. 1 of Part One of the Tariff attached to Presidential Decree 26 April 1986, no. 131 (Approval of the consolidated text of the provisions concerning the registration tax) [...] BELIEVED IN FACT. [...] CONSIDERED IN LAW. [...] The Constitutional Court declares the manifest inadmissibility of the question [...] raised, with reference to the articles 3 and 53 of the Constitution and the principles of reasonableness, rationality, and non-contradiction, by the Provincial Tax Commission of Milan with the ordinance indicated in the epigraph. Thus decided in Rome, in the seat of the Constitutional Court, Palazzo della Consulta, on 22 June 2011.</td><td>ORDER</td></tr>
<tr><td>JUDGMENT TYPE
 
</td><td>In the judgment of constitutional legitimacy of letter b) of number 1) of note II-bis of art. 1 of Part One of the Tariff attached to Presidential Decree 26 April 1986, no. 131 (Approval of the consolidated text of the provisions concerning the registration tax) [...] BELIEVED IN FACT. [...] CONSIDERED IN LAW. [...] The Constitutional Court declares the manifest inadmissibility of the question [...] raised, with reference to the articles 3 and 53 of the Constitution and the principles of reasonableness, rationality, and non-contradiction, by the Provincial Tax Commission of Milan with the ordinance indicated in the epigraph. Thus decided in Rome, in the seat of the Constitutional Court, Palazzo della Consulta, on 22 June 2011.</td><td>JUDGMENT OF CONSTITUTIONAL LEGITIMACY IN AN INCIDENTAL WAY</td></tr>
</table>

FIGURE 6.35. Input-output English example for COMMA tasks.

TABLE 6.55. List of the utilized metrics and their settings.

Metric	Description	Bound†	Hyperparameters
ROUGE-1/2/L [Lin, 2004]	Unigrams (R-1), bigrams (R-2), and longest common subsequence (R-L) lexical overlaps (%).	$[0, 1], \uparrow$	`rouge_types=["rouge1", "rouge2","rougeL"]`, `use_stemmer=True`
$\mathcal{R}$ [Moro et al., 2023b]	Aggregated ROUGE judgment penalizing results with discrepant R-1, R-2, R-L.	$[0, 1], \uparrow$	/
BERTScore [Zhang et al., 2020b]	IDF-weighted n-gram hard-alignment via contextualized BERT embeddings.	$[-1, 1], \uparrow$	`model_type="bert-base-multilingual-cased"`, `rescale_with_baseline=True`, `idf=True`, `batch_size=32`
BARTScore-$\mathcal{F}$ [Yuan et al., 2021]	Multi-perspective evaluation using BART model weights to estimate how likely hypothesis and reference are reciprocal paraphrases. We measure faithfulness as the probability of generating predictions conditioned on the full ruling text (or maxim texts for N → BP).	$[-\infty, 0], \uparrow$	`checkpoint="facebook/mbart-large-cc-25"`, `batch_size=4`
Coverage [Grusky et al., 2018]	The percentage of summary words within the source text. When low, it suggests a high chance of unsupported entities and facts.	$[0, 1], \uparrow$	/
Density [Grusky et al., 2018]	The average character length of extractive fragments. When low, it suggests that most summary sentences are not verbatim extracts from the sources (abstractive).	$[0, x_c], \downarrow^*$	/
Compression [Grusky et al., 2018]	The document-summary word ratio.	$[0, x_w], \uparrow^{**}$	/
Carburacy [Moro et al., 2023b]	Carbon-aware accuracy measure modeling both the generative model effectiveness and eco-sustainability.	$[0, 1], \uparrow$	`alpha=10`, `beta=100`
μ-F_1 [Blair, 1979]	The micro-averaged F1 score aggregates the contributions of all classes to compute the average metric. It gives equal importance to each observation (not indicated for imbalanced classes)	$[0, 1], \uparrow$	/
m-F_1 [Blair, 1979]	The macro-averaged F1 score computes the metric for each class and then takes the average. It gives each class equal importance regardless of their support values.	$[0, 1], \uparrow$	/
NDCG [Järvelin and Kekäläinen, 2002]	The normalized discounted cumulative gain ranks predictions by the given scores, applies a logarithmic discount, and normalizes using the best score value.	$[0, 1], \uparrow$	Given a target article, predictions are awarded scores $\in [0.8, 0.5, 0.2, 0]$ if they refer to the same article, an article from the same Italian Constitution subsection, an article from the same section, or an article from another part, respectively.

* x_c = character-length of the ruling full text (or maxim texts for N → BP).
** x_w = word-length of the ruling full text (or maxim texts for N → BP).
† ↑ = the higher the better, ↓ otherwise.

TABLE 6.56. Zero and few-shot scores per generative task, language, and model. Part 1.

	R-1	R-2	R-L	$\mathcal{R}$	BeS	BaS	Υ_m	t
				AS (r→n)				
IT								
0								
MBART	12.91	4.37	7.53	8.26	-31.22	-3.83	-	510.5
MBART-LSG	27.32	11.68	16.84	18.53	9.47	-7.67	-	148.3
10								
MBART	30.51	14.16	19.40	21.26	5.45	-7.62	50.17	205.1
MBART-LSG	29.82	13.85	18.69	20.69	9.82	-7.52	49.55	164.1
100								
MBART	36.60	18.2	23.19	25.84	11.43	-6.86	54.64	299.6
MBART-LSG	40.32	19.15	23.52	27.43	16.69	-7.33	56.49	279.6
1000								
MBART	39.58	18.92	25.04	27.64	14.16	-7.05	55.18	460.2
MBART-LSG	48.77	28.38	31.73	36.01	26.17	-8.02	61.03	337.8
EN								
0								
MBART	26.46	10.51	16.81	17.85	-17.76	-5.02	-	477.9
MBART-LSG	22.36	10.86	16.45	16.52	2.85	-8.21	-	77.5
10								
MBART	12.94	2.17	12.28	9.11	-51.64	-2.71	34.69	512.0
MBART-LSG	46.72	22.89	27.79	32.13	12.93	-7.43	59.15	319.7
100								
MBART	44.54	22.14	29.83	31.89	7.77	-6.37	59.09	356.0
MBART-LSG	48.70	24.68	29.79	34.03	14.22	-7.58	61.34	337.8
1000								
MBART	49.64	25.23	32.05	35.27	12.68	-5.39	61.46	394.1
MBART-LSG	55.76	31.66	35.92	40.66	22.76	-6.03	64.49	351.3
ES								
0								
MBART	25.12	9.51	15.51	16.64	-21.07	-5.52	-	460.3
MBART-LSG	23.76	12.65	17.63	17.98	5.13	-9.43	-	79.3
10								
MBART	38.22	19.21	25.28	27.40	5.90	-9.13	56.22	192.1
MBART-LSG	45.03	20.91	27.49	30.82	6.43	-8.47	58.51	294.6
100								
MBART	44.71	23.76	28.97	32.22	9.85	-7.95	59.53	251.3
MBART-LSG	50.44	25.46	29.44	34.70	15.48	-8.40	61.82	337.6
1000								
MBART	49.05	24.23	30.92	34.36	10.08	-6.99	60.75	417.0
MBART-LSG	55.81	31.40	34.59	40.13	21.86	-7.66	63.89	363.9
FR								
0								
MBART	26.83	10.26	14.74	17.19	-17.43	-3.68	-	505.5
MBART-LSG	24.88	11.70	16.46	17.63	6.52	-5.89	-	97.1
10								
MBART	34.00	12.96	21.09	22.51	-3.77	-3.89	51.40	471.3
MBART-LSG	48.14	21.70	24.65	31.06	14.82	-4.46	58.13	441.4
100								
MBART	41.09	20.36	25.46	28.75	9.41	-5.15	57.36	284.7
MBART-LSG	48.46	23.71	27.01	32.67	16.36	-5.18	60.19	318.7
1000								
MBART	47.53	23.09	28.37	32.64	11.65	-4.07	60.10	410.8
MBART-LSG	53.72	29.51	31.87	37.92	22.51	-4.18	62.23	370.5

	R-1	R-2	R-L	$\mathcal{R}$	BeS	BaS	Υ_m	t
				AS (r→bp)				
IT								
0								
MBART	9.06	1.88	5.89	5.61	-29.57	-5.06	-	128.0
MBART-LSG	23.14	6.27	13.88	14.36	4.24	-7.77	-	114.6
10								
MBART	20.03	5.90	13.90	13.23	-3.56	-6.38	41.16	154.1
MBART-LSG	20.81	6.09	13.97	13.57	-2.68	-6.98	41.63	148.2
100								
MBART	23.17	9.69	17.05	16.59	12.81	-7.64	45.51	74.3
MBART-LSG	24.62	10.39	18.17	17.67	14.58	-7.52	46.48	86.1
1000								
MBART	34.90	16.72	26.31	25.83	22.72	-8.51	53.60	116.0
MBART-LSG	37.17	18.99	28.51	28.07	26.42	-8.54	54.19	108.1
EN								
0								
MBART	21.90	6.46	15.34	14.51	-19.09	-6.93	-	127.8
MBART-LSG	28.71	9.11	19.32	18.93	-1.63	-8.21	-	77.4
10								
MBART	22.62	7.67	16.91	15.67	-5.32	-8.49	44.59	75.7
MBART-LSG	28.57	10.11	20.51	19.62	-4.32	-7.85	48.81	148.8
100								
MBART	28.56	11.24	21.25	20.25	3.43	-7.30	49.47	142.0
MBART-LSG	31.69	14.15	23.17	22.89	12.23	-8.24	52.12	78.2
1000								
MBART	41.42	19.93	31.06	30.57	19.55	-7.33	58.25	115.6
MBART-LSG	42.66	21.63	32.16	31.91	22.52	-7.41	55.67	108.9
ES								
0								
MBART	22.45	5.72	14.76	14.24	-17.12	-7.47	-	127.8
MBART-LSG	30.41	9.48	19.52	19.66	1.31	-9.43	-	78.3
10								
MBART	23.94	7.66	16.36	15.92	-4.41	-9.81	44.83	59.0
MBART-LSG	26.82	9.74	18.67	18.32	-3.88	-8.94	47.55	73.7
100								
MBART	31.51	12.93	22.61	22.22	7.73	-7.96	51.58	89.2
MBART-LSG	33.10	14.22	23.13	23.34	8.41	-8.73	52.67	99.8
1000								
MBART	40.31	19.29	29.64	29.53	18.36	-8.45	57.22	103.0
MBART-LSG	42.80	22.51	31.93	32.19	23.32	-8.52	57.03	95.0
FR								
0								
MBART	20.96	6.30	12.50	-15.95	13.21	-5.18	-	128.0
MBART-LSG	28.88	9.36	16.77	18.22	2.39	-5.90	-	95.1
10								
MBART	23.71	8.79	15.82	16.05	-0.54	-6.40	44.95	79.1
MBART-LSG	20.84	8.05	13.63	14.13	-0.45	-6.71	42.38	50.9
100								
MBART	28.78	11.82	19.67	19.99	7.44	-5.91	49.16	102.9
MBART-LSG	30.15	12.66	20.17	20.89	10.63	-5.69	50.09	93.6
1000								
MBART	39.23	19.23	27.94	28.61	20.95	-5.67	55.88	113.9
MBART-LSG	37.45	18.44	26.97	27.45	20.51	-5.76	52.89	99.9

TABLE 6.57. Zero and few-shot scores per generative task, language, and model. Part 2.

	R-1	R-2	R-L	$\mathcal{R}$	BeS	BaS	Υ_m	t
AS (n→bp)								
IT								
0								
MBART	23.67	9.35	14.39	15.75	0.02	-5.84	-	114.4
MBART-LSG	28.67	12.26	17.25	19.30	9.63	-6.38	-	101.8
10								
MBART	29.65	12.78	18.55	20.23	7.58	-6.32	49.37	243.6
MBART-LSG	32.20	15.11	19.68	22.21	11.91	-6.66	50.39	300.1
100								
MBART	30.68	14.29	20.81	21.83	13.92	-6.00	51.41	133.1
MBART-LSG	30.53	14.36	20.44	21.68	14.09	-6.30	50.77	103.0
1000								
MBART	43.85	24.18	33.53	33.64	31.42	-6.30	60.26	127.0
MBART-LSG	43.81	23.90	33.34	33.46	31.46	-6.26	60.04	135.3
EN								
0								
MBART	34.61	13.98	21.73	23.27	0.68	-5.08	-	119.6
MBART-LSG	27.84	11.04	18.98	19.20	0.88	-5.69	-	57.8
10								
MBART	33.87	13.76	22.72	23.29	0.80	-4.83	52.78	147.70
MBART-LSG	38.42	17.15	24.24	26.40	7.04	-5.52	55.52	184.4
100								
MBART	33.08	14.87	24.02	23.86	10.38	-5.43	53.04	107.7
MBART-LSG	35.74	15.81	24.15	25.07	6.23	-5.11	53.54	202.1
1000								
MBART	47.35	24.87	35.60	35.64	25.68	-5.40	62.00	122.0
MBART-LSG	47.61	24.72	35.77	35.72	24.60	-5.26	61.66	131.8
ES								
0								
MBART	34.80	12.84	21.25	22.78	2.08	-6.15	-	119.9
MBART-LSG	30.29	11.69	19.74	20.45	4.65	-6.91	-	66.7
10								
MBART	34.93	14.21	22.18	23.60	1.66	-6.13	53.08	132.2
MBART-LSG	36.48	15.13	22.91	24.65	3.02	-6.45	53.41	136.5
100								
MBART	35.06	15.87	24.75	25.07	8.08	-6.14	54.60	113.1
MBART-LSG	37.46	16.90	24.68	26.16	7.78	-6.56	54.65	144.2
1000								
MBART	46.63	24.62	34.50	34.97	24.22	-6.46	61.49	114.4
MBART-LSG	47.44	25.21	35.18	35.65	24.13	-6.39	61.04	118.8
FR								
0								
MBART	32.55	12.76	18.64	21.17	1.91	-3.88	-	122.6
MBART-LSG	27.85	11.14	16.96	18.56	5.04	-4.38	-	69.2
10								
MBART	25.95	10.86	15.85	17.48	-6.21	-4.88	46.61	187.3
MBART-LSG	37.22	15.95	20.98	24.51	8.62	-4.07	53.68	152.8
100								
MBART	30.17	13.79	21.01	21.56	7.86	-3.79	51.19	125.3
MBART-LSG	34.05	15.38	22.12	23.71	9.83	-3.90	52.69	122.7
1000								
MBART	41.54	20.64	28.78	30.10	21.05	-3.98	57.05	132.2
MBART-LSG	45.03	23.36	32.26	33.29	24.89	-3.95	59.77	135.9

	R-1	R-2	R-L	$\mathcal{R}$	BeS	BaS	Υ_m	t
TG								
IT								
0								
MBART	18.41	10.50	13.75	14.21	-16.23	-4.74	-	127.9
MBART-LSG	37.07	23.95	29.89	30.22	24.08	-7.51	-	106.5
10								
MBART	50.08	39.25	44.26	44.44	37.33	-7.74	69.67	141.5
MBART-LSG	46.04	36.89	41.40	41.39	31.34	-7.28	67.96	93.8
100								
MBART	70.27	61.02	65.87	65.63	57.28	-8.29	83.02	134.5
MBART-LSG	62.96	53.86	59.40	58.66	50.89	-7.82	78.52	121.2
1000								
MBART	74.88	66.54	70.74	70.64	63.46	-8.27	84.29	150.7
MBART-LSG	75.15	66.98	71.17	71.02	64.04	-8.27	83.66	149.2
EN								
0								
MBART	36.57	20.84	28.75	28.60	6.36	-7.03	-	127.80
MBART-LSG	37.70	23.54	31.28	30.74	19.91	-8.42	-	70.60
10								
MBART	54.34	40.41	46.56	46.95	35.38	-6.98	71.39	134.80
MBART-LSG	55.35	41.97	49.08	48.65	34.12	-8.21	72.78	134.7
100								
MBART	72.08	61.16	67.43	66.76	56.39	-6.21	83.62	135.6
MBART-LSG	71.75	60.76	67.20	66.43	56.01	-6.25	82.84	131.4
1000								
MBART	74.98	64.70	70.40	69.90	60.39	-6.14	83.93	147.0
MBART-LSG	74.94	64.77	70.27	69.87	60.87	-6.17	83.16	145.3
ES								
0								
MBART	36.39	20.84	27.63	28.17	3.91	-7.37	-	127.3
MBART-LSG	40.58	27.10	33.71	33.69	23.89	-9.44	-	71.5
10								
MBART	46.04	33.08	37.45	38.74	28.16	-7.48	65.69	85.90
MBART-LSG	50.62	38.38	44.67	44.45	32.41	-8.93	69.74	93.0
100								
MBART	70.78	59.06	65.49	64.96	53.06	-7.55	82.65	142.3
MBART-LSG	68.67	57.16	63.25	62.89	50.69	-7.91	80.55	136.5
1000								
MBART	71.26	60.49	66.41	65.93	55.23	-7.63	81.82	135.2
MBART-LSG	73.72	63.06	68.71	68.37	57.71	-7.55	82.31	147.6
FR								
0								
MBART	29.48	17.66	22.00	22.99	-4.56	-3.51	-	128.0
MBART-LSG	37.85	23.97	30.19	30.57	20.79	-5.41	-	83.9
10								
MBART	42.07	31.56	35.62	36.35	26.02	-4.67	63.89	111.3
MBART-LSG	41.21	27.68	32.59	33.72	23.13	-3.88	61.96	226.6
100								
MBART	68.56	57.82	63.33	63.12	52.70	-3.82	81.59	147.3
MBART-LSG	41.88	34.38	39.53	38.56	30.38	-5.13	65.33	147.4
1000								
MBART	73.09	63.51	68.48	68.26	59.91	-3.92	83.01	153.6
MBART-LSG	73.35	63.51	68.53	68.35	59.84	-3.90	82.16	160.1

Recall	
1	Target contents are not covered at all
2	Target contents are partially covered
3	Target contents are fully covered
Precision	
1	The prediction has many contents not covered by the target (unnecessary or redundant information)
2	The prediction contents mostly appear in the target
3	The prediction contains only the target contents.
Faithfulness	
1	The prediction is not factually consistent with the source (hallucinations, polarity inversions, entity misusage)
2	The prediction has few factual inconsistencies w.r.t. source
3	The prediction is factually consistent w.r.t. source

TABLE 6.58. Quality aspect scales and value-level explanations for human evaluation.

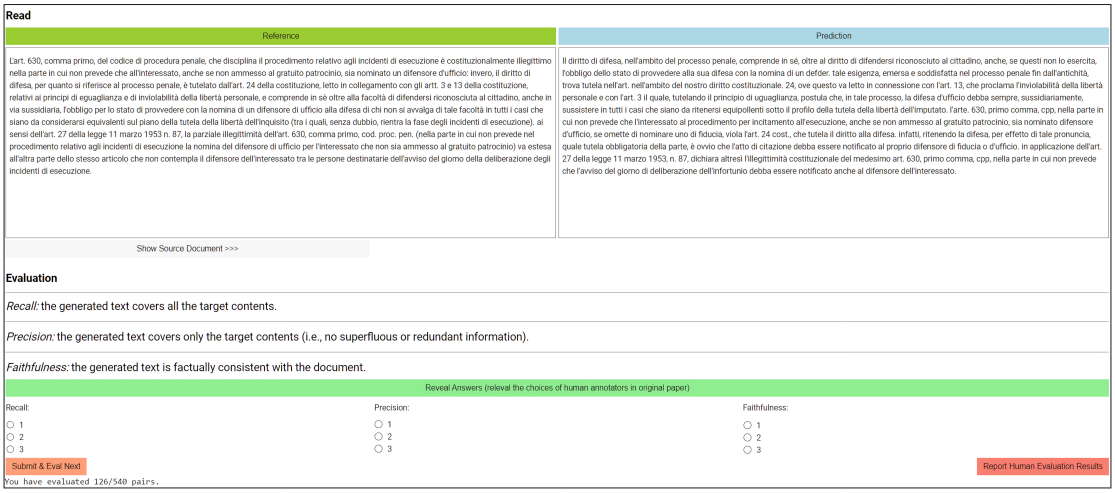


FIGURE 6.36. Screenshot of the human evaluation interface.

TABLE 6.59. CodeCarbon emission statistics with HuggingFace Trainer. Total kg CO_2 in all languages and full/few/zero-shot. Training time.

Task	kg CO_2
AS (r→n)	
MBART	1.93
MBART-LSG	5.63
AS (r→bp)	
MBART	1.95
MBART-LSG	3.09
AS (n→bp)	
MBART	1.79
MBART-LSG	2.59
TG	
MBART	1.71
MBART-LSG	2.86
SLC (Ruling)	
MBART	1.73
XLM-R-LSG	1.90
SLC (Judgment)	
MBART	1.11
XLM-R-LSG	1.38
IR	
XLM-R-LSG	0.4024
MATRIX	0.0220

Chapter 7

Conclusion and Future Work

This thesis delved into the realm of abstractive long-input summarization within the confines of a low-resource environment, characterized by constraints in both data availability and hardware resources. The investigation included various methodological approaches to mitigate the challenges posed by this scenario. In Chapter 3, our exploration began with an examination of text segmentation methods designed to partition large documents into smaller independent segments. Subsequently, in Chapter 4, we implemented techniques that take advantage of a retrieval component to reduce the volume of information, thus directing attention only to the relevant paragraphs. Chapter 5 considered the integration of additional knowledge, using it to infuse and distill insights to enhance model performance. Finally, in Chapter 6, we underscored the pivotal role of novel datasets and benchmarks in incentivizing the application of research findings in real-world scenarios.

This chapter encapsulates the primary discoveries gleaned from the thesis and elucidates prospective avenues for further exploration and research.

7.1 Discussion

Previous work in the domain of long-input summarization focuses primarily on approximating the behavior of the self-attention mechanism [Vaswani et al., 2017]. These efforts propose efficient models characterized by linear time and memory complexity [Beltagy et al., 2020, Huang et al., 2021, Guo et al., 2022a, Phang et al., 2023], facilitating the processing of more extensive inputs. Nevertheless, given that existing solutions still necessitate truncating inputs with excessively long contexts, our attention shifts towards various methods adept at effectively processing entire documents without overlooking any pertinent details. Moreover, although the creation of high-quality datasets demands a substantial investment of time and effort, the efficacy of models is profoundly dependent upon both the quantity and caliber of labeled examples. In this context, previous research has focused mainly on the use of data augmentation techniques to generate additional synthetic samples for training models [Parida and Motlícek, 2019, Magooda and Litman, 2020]. On the contrary, our work establishes the superiority of text segmenta-

tion techniques when applied to lengthy textual inputs [Moro and Ragazzi, 2022, Moro et al., 2022a, 2023d]. By dividing a long source document into smaller, self-contained fragments and aligning them with the more relevant summary portions as labels, we implicitly augment the dataset. Furthermore, this methodology has shown a notable improvement in the focus of the model during training. At the same time, giving the model incremental chunks significantly mitigates the GPU demands required to process an entire long document. This approach provides a memory-efficient solution, particularly beneficial for smaller and medium-sized organizations with limited resources that preclude the acquisition of high-budget GPUs.

Nevertheless, solutions enhanced by segmentation require the model to ingest all of the content, potentially exposing it to considerable noise. In light of this consideration, our research substantiates that a retrieval-enhanced approach possesses the capability to discern crucial information by selectively presenting only those segments to the model. This strategic curation effectively improves performance and reduces the GPU memory requirements that would have been necessary for the query-focused multi-document summarization task [Moro et al., 2022a, 2023e]. However, limiting the solution to summarization based on a user query constrains its applicability in open scenarios. Thus, our study underscores the critical importance of incorporating modules that leverage external knowledge to discern the most important information, improving both model performance and efficiency [Moro et al., 2023c] and interpretability [Moro et al., 2024].

Up to this point, our investigations have focused on small language models (SLMs) boasting million-scale parameters, exemplified by models such as BART and LED. Recent research has supported the increased few-shot capacity of large language models (LLMs) compared to smaller architectures [Brown et al., 2020, Kojima et al., 2022]. However, in practical terms, the computational demands for their deployment remain notably challenging, particularly when dealing with lengthy documents. This challenge arises from the substantial volume of tokens that necessitates processing, leading to prolonged training times and a prerequisite for GPUs with expansive memory capacity. With this consideration in mind, our research delineated the importance of distilling the knowledge from LLMs for applications such as legal question answering [Italiani et al., 2024] and computational fact-checking [Bussotti et al., 2024]. This process is crucial for crafting specialized SLMs, trained on synthetic data generated by powerful larger networks.

Following this direction, we conducted a comprehensive benchmark and analysis to derive overarching recommendations for an eco-sustainable use of models [Moro et al., 2023b]. Notably, we introduced the first accuracy metric specifically designed to assess eco-sustainability, with the intent of catalyzing the development of new efficient state-of-the-art models. Furthermore, our focus extends to the provision of a new dataset, aiming to stimulate research efforts in the real-world domain of multi-task and multi-lingual legal models [Moro et al., 2023a, Ragazzi et al., 2024]. Subsequently, we introduce an extensive study and comparison of decoding strategies for text summarization [Frisoni et al., 2024], a monumental endeavor aimed at providing users with guidance on selecting the optimal strategy based on the specific use case. This research pioneers novel pathways, notably the exploration of differentiable decoding through model emulation.

7.2 Visionary Directions

In light of the substantial influence of LLMs in contemporary contexts and easily accessible services such as ChatGPT [OpenAI, 2023] on mobile devices, their future role is poised to be transformative in diverse domains. These models have exhibited exceptional proficiency in comprehending and generating human-like text, promoting advances in natural language processing [Zhao et al., 2023], information retrieval [Zhu et al., 2023], and even creative content generation [Chen and Ding, 2023, Koh et al., 2023b].

However, several considerations and challenges accompany the proliferation of LLMs: substantial computational requirements, environmental concerns, and potential biases embedded in training data are critical aspects that require meticulous scrutiny. Striking a delicate balance between leveraging the performance capabilities of LLMs and addressing ethical and practical concerns is imperative for their responsible and sustainable integration into future applications [Bai et al., 2024].

Future research efforts should focus on the development of more efficient and environmentally friendly variants of LLMs, addressing resource constraints and fostering accessibility. The ongoing exploration of techniques, including knowledge distillation, model compression, and architectural innovations, will be crucial to enhance the efficiency and applicability of these models across diverse scenarios. Simultaneously, we advocate for future research to explore the integration of our inspected techniques, such as text segmentation, in tandem with state-of-the-art LLMs. The objective is to mitigate the substantial computational overhead associated with these models when processing extensive documents, thereby presenting solutions that are not only efficient and effective but also economically viable for product teams. Further, given that encoder–decoder models currently represent the state-of-the-art in sequence-to-sequence tasks such as text summarization, we recommend the exploration of representational models like Bert integrated with LLMs, as proven successfully in recent studies [Rothe et al., 2020].

Bibliography

The Italian Constitutional Court and Comparative Law. A Premise. 2010.

Amirhossein Abaskohi, Sascha Rothe, and Yadollah Yaghoobzadeh. LM-CPPF: paraphrasing-guided data augmentation for contrastive prompt-based few-shot fine-tuning. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2023, Toronto, Canada, July 9-14, 2023, pages 670–681. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-SHORT.59. URL https://doi.org/10.18653/v1/2023.acl-short.59.

Sahar Abdelnabi, Kai Greshake, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In Maura Pintor, Xinyun Chen, and Florian Tramèr, editors, Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security, AISec 2023, Copenhagen, Denmark, 30 November 2023, pages 79–90. ACM, 2023. doi: 10.1145/3605764.3623985. URL https://doi.org/10.1145/3605764.3623985.

Wasi Uddin Ahmad, Jianfeng Chi, Yuan Tian, and Kai-Wei Chang. Policyqa: A reading comprehension dataset for privacy policies. In Trevor Cohn, Yulan He, and Yang Liu, editors, Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020, volume EMNLP 2020 of Findings of ACL, pages 743–749. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.FINDINGS-EMNLP.66. URL https://doi.org/10.18653/v1/2020.findings-emnlp.66.

Nur Ahmed and Muntasir Wahed. The de-democratization of AI: deep learning and the compute divide in artificial intelligence research. CoRR, abs/2010.15581, 2020. URL https://arxiv.org/abs/2010.15581.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. GQA: training generalized multi-query transformer models from multi-head checkpoints. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023, pages 4895–4901. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.emnlp-main.298.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. FLAIR: an easy-to-use framework for state-of-the-art NLP. In Waleed Ammar, Annie Louis, and Nasrin Mostafazadeh, editors, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations, pages 54–59. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-4010. URL https://doi.org/10.18653/v1/n19-4010.

Suha Al-Thanyyan and Aqil M. Azmi. Automated text simplification: A survey. ACM Comput. Surv., 54(2):43:1–43:36, 2022. doi: 10.1145/3442695. URL https://doi.org/10.1145/3442695.

Eman Rashed Alkatheery. Google translate errors in legal texts: Machine translation quality assessment. SocArXiv. March, 15, 2023.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. The falcon series of open language models. CoRR, abs/2311.16867, 2023. doi: 10.48550/ARXIV.2311.16867. URL https://doi.org/10.48550/arXiv.2311.16867.

Nouf Ibrahim Altmami and Mohamed El Bachir Menai. Automatic summarization of scientific articles: A survey. J. King Saud Univ. Comput. Inf. Sci., 34(4):1011–1028, 2022. doi: 10.1016/J.JKSUCI.2020.04.020. URL https://doi.org/10.1016/j.jksuci.2020.04.020.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. FEVEROUS: fact extraction and verification over unstructured and structured information. In Joaquin Vanschoren and Sai-Kit Yeung, editors, Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/68d30a9594728bc39aa24be94b319d21-Abstract-round1.html.

Reinald Kim Amplayo and Mirella Lapata. Informative and controllable opinion summarization. In Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty, editors, Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021, pages 2662–2672. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EACL-MAIN.229. URL https://doi.org/10.18653/v1/2021.eacl-main.229.

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. Do not have enough data? deep learning to the rescue! In The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 7383–7390. AAAI Press, 2020. doi: 10.1609/AAAI.V34I05.6233. URL https://doi.org/10.1609/aaai.v34i05.6233.

Deepa Anand and Rupali S. Wagh. Effective deep learning approaches for summarization of legal texts. J. King Saud Univ. Comput. Inf. Sci., 34(5):2141–2150, 2022. doi: 10.1016/J.JKSUCI.2019.11.015. URL https://doi.org/10.1016/j.jksuci.2019.11.015.

Lasse F. Wolff Anthony, Benjamin Kanding, and Raghavendra Selvan. Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. CoRR, abs/2007.03051, 2020. URL https://arxiv.org/abs/2007.03051.

Diego Antognini and Boi Faltings. Learning to create sentence semantic relation graphs for multi-document summarization. CoRR, abs/1909.12231, 2019. URL http://arxiv.org/abs/1909.12231.

Rahul Aralikatte, Shashi Narayan, Joshua Maynez, Sascha Rothe, and Ryan T. McDonald. Focus attention: Promoting faithfulness and diversity in summarization. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, pages 6078–6095. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.ACL-LONG.474. URL https://doi.org/10.18653/v1/2021.acl-long.474.

Dennis Aumiller, Satya Almasian, Sebastian Lackner, and Michael Gertz. Structural text segmentation of legal documents. In Juliano Maranhão and Adam Zachary Wyner, editors, ICAIL '21: Eighteenth International Conference for Artificial Intelligence and Law, São Paulo Brazil, June 21 - 25, 2021, pages 2–11. ACM, 2021. doi: 10.1145/3462757.3466085. URL https://doi.org/10.1145/3462757.3466085.

Dennis Aumiller, Ashish Chouhan, and Michael Gertz. Eur-lex-sum: A multi- and cross-lingual dataset for long-form summarization in the legal domain. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, pages 7626–7639. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.519. URL https://doi.org/10.18653/v1/2022.emnlp-main.519.

Sanghwan Bae, Taeuk Kim, Jihoon Kim, and Sang-goo Lee. Summary level training of sentence rewriting for abstractive summarization. CoRR, abs/1909.08752, 2019. URL http://arxiv.org/abs/1909.08752.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015. URL http://arxiv.org/abs/1409.0473.

Guangji Bai, Zheng Chai, Chen Ling, Shiyu Wang, Jiaying Lu, Nan Zhang, Tingwei Shi, Ziyang Yu, Mengdan Zhu, Yifei Zhang, Carl J. Yang, Yue Cheng, and Liang Zhao. Beyond efficiency: A systematic survey of resource-efficient large language models. CoRR, abs/2401.00625, 2024. doi: 10.48550/ARXIV.2401.00625. URL https://doi.org/10.48550/arXiv.2401.00625.

Ahsaas Bajaj, Pavitra Dangati, Kalpesh Krishna, Pradhiksha Ashok Kumar, Rheeya Uppaal, Bradford Windsor, Eliot Brenner, Dominic Dotterrer, Rajarshi Das, and Andrew McCallum. Long document summarization in a low resource setting using pretrained language models. In Jad Kabbara, Haitao Lin, Amandalynne Paullada, and Jannis Vamvas, editors, Proceedings of the ACL-IJCNLP 2021 Student Research Workshop, ACL 2021, Online, JUli 5-10, 2021, pages 71–80. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.ACL-SRW.7. URL https://doi.org/10.18653/v1/2021.acl-srw.7.

Nesrine Bannour, Sahar Ghannay, Aurélie Névéol, and Anne-Laure Ligozat. Evaluating the carbon footprint of NLP methods: a survey and analysis of existing tools. In Nafise Sadat Moosavi, Iryna Gurevych, Angela Fan, Thomas Wolf, Yufang Hou, Ana Marasovic, and Sujith Ravi, editors, Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing, SustaiNLP@EMNLP 2021, Virtual, November 10, 2021, pages 11–21. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.SUSTAINLP-1.2. URL https://doi.org/10.18653/v1/2021.sustainlp-1.2.

Roy Bar-Haim, Lilach Eden, Roni Friedman, Yoav Kantor, Dan Lahav, and Noam Slonim. From arguments to key points: Towards automatic argument summarization. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pages 4029–4039. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.ACL-MAIN.371. URL https://doi.org/10.18653/v1/2020.acl-main.371.

Alvaro Bartolome, Gabriel Martin, and Daniel Vila. Notus. https://github.com/argilla-io/notus, 2023.

Sourya Basu, Govardana Sachitanandam Ramachandran, Nitish Shirish Keskar, and Lav R. Varshney. Mirostat: a neural text decoding algorithm that directly controls perplexity. In 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net, 2021. URL https://openreview.net/forum?id=W1G1JZEIy5_.

Valerio Bellandi, Silvana Castano, Paolo Ceravolo, Ernesto Damiani, Alfio Ferrara, Stefano Montanelli, Sergio Picascia, Antongiacomo Polimeno, and Davide Riva. Knowledge-based legal document retrieval: A case study on italian civil court decisions. In Danai Symeonidou, Ran Yu, Davide Ceolin, María Poveda-Villalón, Davide Audrito, Luigi Di Caro, Francesca Grasso, Roberto Nai, Emilio Sulis, Fajar J. Ekaputra, Oliver Kutz, and Nicolas Troquard, editors, Companion Proceedings of the 23rd International Conference on Knowledge Engineering and Knowledge Management, Bozen-Bolzano, Italy, September 26-29, 2022, volume 3256 of CEUR Workshop Proceedings. CEUR-WS.org, 2022. URL https://ceur-ws.org/Vol-3256/km4law2.pdf.

Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pages 3613–3618. Association for Computational Linguistics, 2019. doi: 10.18653/V1/D19-1371. URL https://doi.org/10.18653/v1/D19-1371.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. CoRR, abs/2004.05150, 2020. URL https://arxiv.org/abs/2004.05150.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In Madeleine Clare Elish, William Isaac, and Richard S. Zemel, editors, FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021, pages 610–623. ACM, 2021. doi: 10.1145/3442188.3445922. URL https://doi.org/10.1145/3442188.3445922.

William Berrios, Gautam Mittal, Tristan Thrush, Douwe Kiela, and Amanpreet Singh. Towards language models that can see: Computer vision through the LENS of natural language. CoRR, abs/2306.16410, 2023. doi: 10.48550/ARXIV.2306.16410. URL https://doi.org/10.48550/arXiv.2306.16410.

Rupal Bhargava, Sukrut Nigwekar, and Yashvardhan Sharma. Catchphrase extraction from legal documents using LSTM networks. In Prasenjit Majumder, Mandar Mitra, Parth Mehta, and Jainisha Sankhavara, editors, Working notes of FIRE 2017 - Forum for Information Retrieval Evaluation, Bangalore, India, December 8-10, 2017, volume 2036 of CEUR Workshop Proceedings, pages 72–73. CEUR-WS.org, 2017. URL http://ceur-ws.org/Vol-2036/T3-3.pdf.

Paheli Bhattacharya, Kripabandhu Ghosh, Saptarshi Ghosh, Arindam Pal, Parth Mehta, Arnab Bhattacharya, and Prasenjit Majumder. FIRE 2019 AILA track: Artificial intelligence for legal assistance. In Prasenjit Majumder, Mandar Mitra, Surupendu Gangopadhyay, and Parth Mehta, editors, FIRE '19: Forum for Information Retrieval Evaluation, Kolkata, India, December, 2019, pages 4–6. ACM, 2019. doi: 10.1145/3368567.3368587. URL https://doi.org/10.1145/3368567.3368587.

Steven Bird. NLTK: the natural language toolkit. In Nicoletta Calzolari, Claire Cardie, and Pierre Isabelle, editors, ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006. The Association for Computer Linguistics, 2006. doi: 10.3115/1225403.1225421. URL https://aclanthology.org/P06-4018/.

David C. Blair. Information Retrieval, 2nd ed. C.J. Van Rijsbergen. London: Butterworths. Journal of the American Society for Information Science, 30(6):374–375, November 1979. doi: 10.1002/asi.4630300621.

Leonard Bongard, Lena Held, and Ivan Habernal. The legal argument reasoning task in civil procedure. In Nikolaos Aletras, Ilias Chalkidis, Leslie Barrett, Catalina Goanta, and Daniel Preotiuc-Pietro, editors, Proceedings of the Natural Legal Language Processing Workshop, NLLP@EMNLP 2022, Abu Dhabi, United Arab Emirates (Hybrid), December 8, 2022, pages 194–207. Association for Computational Linguistics, 2022. URL https://aclanthology.org/2022.nllp-1.17.

Luiz Henrique Bonifacio, Hugo Queiroz Abonizio, Marzieh Fadaee, and Rodrigo Frassetto Nogueira. Inpars: Data augmentation for information retrieval using large language models. CoRR, abs/2202.05144, 2022. URL https://arxiv.org/abs/2202.05144.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA, volume 162 of Proceedings of Machine Learning Research, pages 2206–2240. PMLR, 2022. URL https://proceedings.mlr.press/v162/borgeaud22a.html.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Jean-Flavien Bussotti, Enzo Veltri, Donatello Santoro, and Paolo Papotti. Generation of training examples for tabular natural language inference. Proc. ACM Manag. Data, 1(4):243:1–243:27, 2023. doi: 10.1145/3626730. URL https://doi.org/10.1145/3626730.

Jean Flavien Bussotti, Luca Ragazzi, Giacomo Frisoni, Gianluca Moro, and Paolo Papotti. Un*kn*own claims: Generation of fact-checking training examples from unstructured and structured data, 2024. Submitted to the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024.

Massimo Caccia, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin. Language gans falling short. In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020. URL https://openreview.net/forum?id=BJgza6VtPB.

Jiahang Cao, Jinyuan Fang, Zaiqiao Meng, and Shangsong Liang. Knowledge graph embedding: A survey from the perspective of representation spaces. CoRR, abs/2211.03536, 2022a. doi: 10.48550/ARXIV.2211.03536. URL https://doi.org/10.48550/arXiv.2211.03536.

Qingqing Cao, Aruna Balasubramanian, and Niranjan Balasubramanian. Towards accurate and reliable energy measurement of NLP models. In Nafise Sadat Moosavi, Angela Fan, Vered Shwartz, Goran Glavas, Shafiq R. Joty, Alex Wang, and Thomas Wolf, editors, Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing, SustaiNLP@EMNLP 2020, Online, November 20, 2020, pages 141–148. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.SUSTAINLP-1.19. URL https://doi.org/10.18653/v1/2020.sustainlp-1.19.

Qingqing Cao, Yash Kumar Lal, Harsh Trivedi, Aruna Balasubramanian, and Niranjan Balasubramanian. Irene: Interpretable energy prediction for transformers. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, pages

2145–2157. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.ACL-LONG.167. URL https://doi.org/10.18653/v1/2021.acl-long.167.

Shuyang Cao and Lu Wang. HIBRIDS: attention with hierarchical biases for structure-aware long document summarization. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 786–807. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.ACL-LONG.58. URL https://doi.org/10.18653/v1/2022.acl-long.58.

Yu Cao, Wei Bi, Meng Fang, Shuming Shi, and Dacheng Tao. A model-agnostic data manipulation method for persona-based dialogue generation. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 7984–8002. Association for Computational Linguistics, 2022b. doi: 10.18653/V1/2022.ACL-LONG.550. URL https://doi.org/10.18653/v1/2022.acl-long.550.

Kevin Matthe Caramancion. News verifiers showdown: A comparative performance evaluation of chatgpt 3.5, chatgpt 4.0, bing ai, and bard in news fact-checking. CoRR, abs/2306.17176, 2023. doi: 10.48550/ARXIV.2306.17176. URL https://doi.org/10.48550/arXiv.2306.17176.

John M Carey, Andrew M Guess, Peter J Loewen, Eric Merkley, Brendan Nyhan, Joseph B Phillips, and Jason Reifler. The ephemeral effects of fact-checks on covid-19 misperceptions in the united states, great britain and canada. Nature Human Behaviour, 6(2):236–243, 2022.

Aldo Gael Carranza, Rezsa Farahani, Natalia Ponomareva, Alex Kurakin, Matthew Jagielski, and Milad Nasr. Privacy-preserving recommender systems with synthetic query generation using differentially private large language models. CoRR, abs/2305.05973, 2023. doi: 10.48550/ARXIV.2305.05973. URL https://doi.org/10.48550/arXiv.2305.05973.

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. Deep communicating agents for abstractive summarization. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers), pages 1662–1675. Association for Computational Linguistics, 2018. doi: 10.18653/V1/N18-1150. URL https://doi.org/10.18653/v1/n18-1150.

Walter Cerroni, Gianluca Moro, Tommaso Pirini, and Marco Ramilli. Peer-to-peer data mining classifiers for decentralized detection of network attacks. In ADC, volume 137 of CRPIT, pages 101–108. ACS, 2013.

Walter Cerroni, Gianluca Moro, Roberto Pasolini, and Marco Ramilli. Decentralized detection of network attacks through P2P data clustering of SNMP data. Comput. Secur., 52:1–16, 2015. doi: 10.1016/J.COSE.2015.03.006. URL https://doi.org/10.1016/j.cose.2015.03.006.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. LEGAL-BERT: The muppets straight out of law school. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 2898–2904, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.261. URL https://aclanthology.org/2020.findings-emnlp.261.

Ilias Chalkidis, Manos Fergadiotis, and Ion Androutsopoulos. Multieurlex - A multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021,

Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, pages 6974–6996. Association for Computational Linguistics, 2021a. doi: 10.18653/V1/2021.EMNLP-MAIN.559. URL https://doi.org/10.18653/v1/2021.emnlp-main.559.

Ilias Chalkidis, Manos Fergadiotis, Nikolaos Manginas, Eva Katakalou, and Prodromos Malakasiotis. Regulatory compliance through doc2doc information retrieval: A case study in EU/UK legislation where text similarity has limitations. In Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty, editors, Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021, pages 3498–3511. Association for Computational Linguistics, 2021b. doi: 10.18653/V1/2021.EACL-MAIN.305. URL https://doi.org/10.18653/v1/2021.eacl-main.305.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael J. Bommarito II, Ion Androutsopoulos, Daniel Martin Katz, and Nikolaos Aletras. Lexglue: A benchmark dataset for legal language understanding in english. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 4310–4330. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.ACL-LONG.297. URL https://doi.org/10.18653/v1/2022.acl-long.297.

Ilias Chalkidis, Nicolas Garneau, Catalina Goanta, Daniel Martin Katz, and Anders Søgaard. Lexfiles and legallama: Facilitating english multinational legal language model development. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023, pages 15513–15535. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.865. URL https://doi.org/10.18653/v1/2023.acl-long.865.

Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. Booookscore: A systematic exploration of book-length summarization in the era of llms. CoRR, abs/2310.00785, 2023. doi: 10.48550/ARXIV.2310.00785. URL https://doi.org/10.48550/arXiv.2310.00785.

Dongmei Chen, Sheng Zhang, Xin Zhang, and Kaijing Yang. Cross-lingual passage re-ranking with alignment augmented multilingual BERT. IEEE Access, 8:213232–213243, 2020a. doi: 10.1109/ACCESS.2020.3041605. URL https://doi.org/10.1109/ACCESS.2020.3041605.

Guanzheng Chen, Jinyuan Fang, Zaiqiao Meng, Qiang Zhang, and Shangsong Liang. Multi-relational graph representation learning with bayesian gaussian process network. In Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022, pages 5530–5538. AAAI Press, 2022a. doi: 10.1609/AAAI.V36I5.20492. URL https://doi.org/10.1609/aaai.v36i5.20492.

Honghua Chen and Nai Ding. Probing the "creativity" of large language models: Can models produce divergent semantic association? In Houda Bouamor, Juan Pino, and Kalika Bali, editors, Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023, pages 12881–12888. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.findings-emnlp.858.

Moye Chen, Wei Li, Jiachen Liu, Xinyan Xiao, Hua Wu, and Haifeng Wang. Sgsum: Transforming multi-document summarization into sub-graph selection. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, pages 4063–4074. Association for Computational Linguistics, 2021a. doi: 10.18653/V1/2021.EMNLP-MAIN.333. URL https://doi.org/10.18653/v1/2021.emnlp-main.333.

Qingyu Chen, Alexis Allot, and Zhiyong Lu. Litcovid: an open database of COVID-19 literature. Nucleic Acids Res., 49(Database-Issue):D1534–D1540, 2021b. doi: 10.1093/NAR/GKAA952. URL https://doi.org/10.1093/nar/gkaa952.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. Tabfact: A large-scale dataset for table-based fact verification. In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020b. URL https://openreview.net/forum?id=rkeJRhNYDH.

Xiuying Chen, Hind Alamro, Mingzhe Li, Shen Gao, Xiangliang Zhang, Dongyan Zhao, and Rui Yan. Capturing relations between scientific papers: An abstractive model for related work section generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, pages 6068–6077. Association for Computational Linguistics, 2021c. doi: 10.18653/V1/2021.ACL-LONG.473. URL https://doi.org/10.18653/v1/2021.acl-long.473.

Yen-Chun Chen, Zhe Gan, Yu Cheng, Jingzhou Liu, and Jingjing Liu. Distilling the knowledge of BERT for text generation. CoRR, abs/1911.03829, 2019. URL http://arxiv.org/abs/1911.03829.

Yi-Syuan Chen and Hong-Han Shuai. Meta-transfer learning for low-resource abstractive summarization. In Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021, pages 12692–12700. AAAI Press, 2021. doi: 10.1609/AAAI.V35I14.17503. URL https://doi.org/10.1609/aaai.v35i14.17503.

Yutian Chen, Xingyou Song, Chansoo Lee, Zi Wang, Richard Zhang, David Dohan, Kazuya Kawakami, Greg Kochanski, Arnaud Doucet, Marc'Aurelio Ranzato, Sagi Perel, and Nando de Freitas. Towards learning universal hyperparameter optimizers with transformers. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022, 2022b. URL http://papers.nips.cc/paper_files/paper/2022/hash/cf6501108fced72ee5c47e2151c4e153-Abstract-Conference.html.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 1724–1734. ACL, 2014. doi: 10.3115/V1/D14-1179. URL https://doi.org/10.3115/v1/d14-1179.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J. Colwell, and Adrian Weller. Rethinking attention with performers. In 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net, 2021. URL https://openreview.net/forum?id=Ua6zuk0WRH.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny

Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. J. Mach. Learn. Res., 24:240:1–240:113, 2023. URL http://jmlr.org/papers/v24/22-1144.html.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. CoRR, abs/2210.11416, 2022. doi: 10.48550/ARXIV.2210.11416. URL https://doi.org/10.48550/arXiv.2210.11416.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: pre-training text encoders as discriminators rather than generators. In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020. URL https://openreview.net/forum?id=r1xMH1BtvB.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers), pages 615–621. Association for Computational Linguistics, 2018. doi: 10.18653/V1/N18-2097. URL https://doi.org/10.18653/v1/n18-2097.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. SPECTER: document-level representation learning using citation-informed transformers. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pages 2270–2282. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.ACL-MAIN.207. URL https://doi.org/10.18653/v1/2020.acl-main.207.

Ronan Collobert, Awni Y. Hannun, and Gabriel Synnaeve. A fully differentiable beam search decoder. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, volume 97 of Proceedings of Machine Learning Research, pages 1341–1350. PMLR, 2019. URL http://proceedings.mlr.press/v97/collobert19a.html.

European Commission. On artificial intelligence—a european approach to excellence and trust, 2020.

Charles Condevaux and Sébastien Harispe. LSG attention: Extrapolation of pretrained transformers to long sequences. In Hisashi Kashima, Tsuyoshi Idé, and Wen-Chih Peng, editors, Advances in Knowledge Discovery and Data Mining - 27th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2023, Osaka, Japan, May 25-28, 2023, Proceedings, Part I, volume 13935 of Lecture Notes in Computer Science, pages 443–454. Springer, 2023. doi: 10.1007/978-3-031-33374-3_35. URL https://doi.org/10.1007/978-3-031-33374-3_35.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, Proceedings of the 58th Annual Meeting of the Association for Computational

Linguistics, ACL 2020, Online, July 5-10, 2020, pages 8440–8451. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.ACL-MAIN.747. URL https://doi.org/10.18653/v1/2020.acl-main.747.

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. No language left behind: Scaling human-centered machine translation. CoRR, abs/2207.04672, 2022. doi: 10.48550/ARXIV.2207.04672. URL https://doi.org/10.48550/arXiv.2207.04672.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. Overview of the TREC 2019 deep learning track. CoRR, abs/2003.07820, 2020. URL https://arxiv.org/abs/2003.07820.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with high-quality feedback. CoRR, abs/2310.01377, 2023. doi: 10.48550/ARXIV.2310.01377. URL https://doi.org/10.48550/arXiv.2310.01377.

Peng Cui and Le Hu. Sliding selector network with dynamic memory for extractive summarization of long documents. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021, pages 5881–5891. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.NAACL-MAIN.470. URL https://doi.org/10.18653/v1/2021.naacl-main.470.

Josenildo Costa da Silva, Matthias Klusch, Stefano Lodi, and Gianluca Moro. Privacy-preserving agent-based distributed data clustering. Web Intell. Agent Syst., 4(2):221–238, 2006. URL http://content.iospress.com/articles/web-intelligence-and-agent-systems-an-international-journal/wia00088.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL recognising textual entailment challenge. In Joaquin Quiñonero Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alché-Buc, editors, Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers, volume 3944 of Lecture Notes in Computer Science, pages 177–190. Springer, 2005. doi: 10.1007/11736790_9. URL https://doi.org/10.1007/11736790_9.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pages 2978–2988. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1285. URL https://doi.org/10.18653/v1/p19-1285.

Mayukh Das and Wolf-Tilo Balke. Quantifying bias from decoding techniques in natural language generation. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee,

Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022, pages 1311–1323. International Committee on Computational Linguistics, 2022. URL https://aclanthology.org/2022.coling-1.112.

Ona de Gibert Bonet, Aitor García Pablos, Montse Cuadros, and Maite Melero. Spanish datasets for sensitive entity detection in the legal domain. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Jan Odijk, and Stelios Piperidis, editors, Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022, Marseille, France, 20-25 June 2022, pages 3751–3760. European Language Resources Association, 2022. URL https://aclanthology.org/2022.lrec-1.400.

Diego de Vargas Feijó and Viviane Pereira Moreira. Summarizing legal rulings: Comparative experiments. In Ruslan Mitkov and Galia Angelova, editors, Proceedings of the International Conference on Recent Advances in Natural Language Processing, RANLP 2019, Varna, Bulgaria, September 2-4, 2019, pages 313–322. INCOMA Ltd., 2019. doi: 10.26615/978-954-452-056-4_036. URL https://doi.org/10.26615/978-954-452-056-4_036.

Alexandra DeLucia, Aaron Mueller, Xiang Lisa Li, and João Sedoc. Decoding methods for neural narrative generation. CoRR, abs/2010.07375, 2020. URL https://arxiv.org/abs/2010.07375.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1423. URL https://doi.org/10.18653/v1/n19-1423.

Alvin Dey, Tanya Chowdhury, Yash Kumar Atri, and Tanmoy Chakraborty. Corpora evaluation and system bias detection in multi document summarization. In Trevor Cohn, Yulan He, and Yang Liu, editors, Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020, volume EMNLP 2020 of Findings of ACL, pages 2830–2840. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.FINDINGS-EMNLP.254. URL https://doi.org/10.18653/v1/2020.findings-emnlp.254.

Jay DeYoung, Iz Beltagy, Madeleine van Zuylen, Bailey Kuehl, and Lucy Lu Wang. Ms\^2: Multi-document summarization of medical studies. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, pages 7494–7513. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EMNLP-MAIN.594. URL https://doi.org/10.18653/v1/2021.emnlp-main.594.

Payal Dhar. The carbon impact of artificial intelligence. Nat. Mach. Intell., 2(8):423–425, 2020. doi: 10.1038/S42256-020-0219-9. URL https://doi.org/10.1038/s42256-020-0219-9.

Pietro di Lena, Giacomo Domeniconi, Luciano Margara, and Gianluca Moro. GOTA: GO term annotation of biomedical literature. BMC Bioinform., 16:346:1–346:13, 2015. doi: 10.1186/S12859-015-0777-8. URL https://doi.org/10.1186/s12859-015-0777-8.

Siyu Ding, Junyuan Shang, Shuohuan Wang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. Ernie-doc: A retrospective long-document modeling transformer. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language

Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, pages 2914–2927. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.ACL-LONG.227. URL https://doi.org/10.18653/v1/2021.acl-long.227.

Xuan-Dung Doan, Le-Minh Nguyen, and Khac-Hoai Nam Bui. Multi graph neural network for extractive long document summarization. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022, pages 5870–5875. International Committee on Computational Linguistics, 2022. URL https://aclanthology.org/2022.coling-1.512.

Radina Dobreva and Frank Keller. Investigating negation in pre-trained vision-and-language models. In Jasmijn Bastings, Yonatan Belinkov, Emmanuel Dupoux, Mario Giulianelli, Dieuwke Hupkes, Yuval Pinter, and Hassan Sajjad, editors, Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2021, Punta Cana, Dominican Republic, November 11, 2021, pages 350–362. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.BLACKBOXNLP-1.27. URL https://doi.org/10.18653/v1/2021.blackboxnlp-1.27.

Giacomo Domeniconi, Marco Masseroli, Gianluca Moro, and Pietro Pinoli. Discovering new gene functionalities from random perturbations of known gene ontological annotations. In Ana L. N. Fred and Joaquim Filipe, editors, KDIR 2014 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, Rome, Italy, 21 - 24 October, 2014, pages 107–116. SciTePress, 2014a. doi: 10.5220/0005087801070116. URL https://doi.org/10.5220/0005087801070116.

Giacomo Domeniconi, Gianluca Moro, Roberto Pasolini, and Claudio Sartori. Cross-domain text classification through iterative refining of target categories representations. In Ana L. N. Fred and Joaquim Filipe, editors, KDIR 2014 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, Rome, Italy, 21 - 24 October, 2014, pages 31–42. SciTePress, 2014b. doi: 10.5220/0005069400310042. URL https://doi.org/10.5220/0005069400310042.

Giacomo Domeniconi, Gianluca Moro, Roberto Pasolini, and Claudio Sartori. Iterative refining of category profiles for nearest centroid cross-domain text classification. In Ana L. N. Fred, Jan L. G. Dietz, David Aveiro, Kecheng Liu, and Joaquim Filipe, editors, Knowledge Discovery, Knowledge Engineering and Knowledge Management - 6th International Joint Conference, IC3K 2014, Rome, Italy, October 21-24, 2014, Revised Selected Papers, volume 553 of Communications in Computer and Information Science, pages 50–67. Springer, 2014c. doi: 10.1007/978-3-319-25840-9_4. URL https://doi.org/10.1007/978-3-319-25840-9_4.

Giacomo Domeniconi, Gianluca Moro, Andrea Pagliarani, and Roberto Pasolini. Markov chain based method for in-domain and cross-domain sentiment classification. In Ana L. N. Fred, Jan L. G. Dietz, David Aveiro, Kecheng Liu, and Joaquim Filipe, editors, KDIR 2015 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, part of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015), Volume 1, Lisbon, Portugal, November 12-14, 2015, pages 127–137. SciTePress, 2015a. doi: 10.5220/0005636001270137. URL https://doi.org/10.5220/0005636001270137.

Giacomo Domeniconi, Gianluca Moro, Roberto Pasolini, and Claudio Sartori. A comparison of term weighting schemes for text classification and sentiment analysis with a supervised variant of tf.idf. In Markus Helfert, Andreas Holzinger, Orlando Belo, and Chiara Francalanci, editors, Data Management Technologies and Applications - 4th International Conference, DATA 2015, Colmar, France, July 20-22, 2015, Revised Selected Papers, volume 584 of Communications in Computer and Information Science, pages 39–58. Springer, 2015b. doi: 10.1007/978-3-319-30162-4_4. URL https://doi.org/10.1007/978-3-319-30162-4_4.

Giacomo Domeniconi, Marco Masseroli, Gianluca Moro, and Pietro Pinoli. Cross-organism learning method to discover new gene functionalities. Comput. Methods Programs Biomed., 126:20–34, 2016a. doi: 10.1016/J.CMPB.2015.12.002. URL https://doi.org/10.1016/j.cmpb.2015.12.002.

Giacomo Domeniconi, Gianluca Moro, Andrea Pagliarani, Karin Pasini, and Roberto Pasolini. Job recommendation from semantic similarity of linkedin users' skills. In Maria De Marsico, Gabriella Sanniti di Baja, and Ana L. N. Fred, editors, Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods, ICPRAM 2016, Rome, Italy, February 24-26, 2016, pages 270–277. SciTePress, 2016b. doi: 10.5220/0005702302700277. URL https://doi.org/10.5220/0005702302700277.

Giacomo Domeniconi, Konstantinos Semertzidis, Vanessa López, Elizabeth M. Daly, Spyros Kotoulas, and Gianluca Moro. A novel method for unsupervised and supervised conversational message thread detection. In Chiara Francalanci and Markus Helfert, editors, DATA 2016 - Proceedings of 5th International Conference on Data Management Technologies and Applications, Lisbon, Portugal, 24-26 July, 2016, pages 43–54. SciTePress, 2016c. doi: 10.5220/0006001100430054. URL https://doi.org/10.5220/0006001100430054.

Giacomo Domeniconi, Gianluca Moro, Andrea Pagliarani, and Roberto Pasolini. On deep learning in cross-domain sentiment classification. In Ana L. N. Fred and Joaquim Filipe, editors, Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - (Volume 1), Funchal, Madeira, Portugal, November 1-3, 2017, pages 50–60. SciTePress, 2017. doi: 10.5220/0006488100500060. URL https://doi.org/10.5220/0006488100500060.

Yue Dong, Yikang Shen, Eric Crawford, Herke van Hoof, and Jackie Chi Kit Cheung. Banditsum: Extractive summarization as a contextual bandit. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, pages 3739–3748. Association for Computational Linguistics, 2018. doi: 10.18653/v1/d18-1409. URL https://doi.org/10.18653/v1/d18-1409.

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten P. Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen S. Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V. Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. Glam: Efficient scaling of language models with mixture-of-experts. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA, volume 162 of Proceedings of Machine Learning Research, pages 5547–5569. PMLR, 2022. URL https://proceedings.mlr.press/v162/du22c.html.

Bryan Eikema and Wilker Aziz. Is MAP decoding all you need? the inadequacy of the mode in neural machine translation. In Donia Scott, Núria Bel, and Chengqing Zong, editors, Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020, pages 4506–4520. International Committee on Computational Linguistics, 2020. doi: 10.18653/V1/2020.COLING-MAIN.398. URL https://doi.org/10.18653/v1/2020.coling-main.398.

Liat Ein-Dor, Yosi Mass, Alon Halfon, Elad Venezian, Ilya Shnayderman, Ranit Aharonov, and Noam Slonim. Learning thematic similarity metric from article sections using triplet networks. In Iryna Gurevych and Yusuke Miyao, editors, Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers, pages 49–54. Association for Computational Linguistics, 2018. doi: 10.18653/V1/P18-2009. URL https://aclanthology.org/P18-2009/.

Denis Emelin, Daniele Bonadiman, Sawsan Alqahtani, Yi Zhang, and Saab Mansour. Injecting domain knowledge in language models for task-oriented dialogue systems. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, pages 11962–11974. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.820. URL https://doi.org/10.18653/v1/2022.emnlp-main.820.

Alexander R. Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R. Radev. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pages 1074–1084. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1102. URL https://doi.org/10.18653/v1/p19-1102.

Alexander R. Fabbri, Wojciech Kryscinski, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir R. Radev. Summeval: Re-evaluating summarization evaluation. Trans. Assoc. Comput. Linguistics, 9:391–409, 2021. doi: 10.1162/TACL_A_00373. URL https://doi.org/10.1162/tacl_a_00373.

Angela Fan, Mike Lewis, and Yann N. Dauphin. Hierarchical neural story generation. In Iryna Gurevych and Yusuke Miyao, editors, Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers, pages 889–898. Association for Computational Linguistics, 2018. doi: 10.18653/V1/P18-1082. URL https://aclanthology.org/P18-1082/.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. ELI5: long form question answering. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pages 3558–3567. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1346. URL https://doi.org/10.18653/v1/p19-1346.

Xiachong Feng, Xiaocheng Feng, and Bing Qin. Msamsum: Towards benchmarking multi-lingual dialogue summarization. In Song Feng, Hui Wan, Caixia Yuan, and Han Yu, editors, Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering, DialDoc@ACL 2022, Dublin, Ireland, May 26, 2022, pages 1–12. Association for Computational Linguistics, 2022a. doi: 10.18653/V1/2022.DIALDOC-1.1. URL https://doi.org/10.18653/v1/2022.dialdoc-1.1.

Xiachong Feng, Xiaocheng Feng, and Bing Qin. A survey on dialogue summarization: Recent advances and new frontiers. In Luc De Raedt, editor, Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022, pages 5453–5460. ijcai.org, 2022b. doi: 10.24963/ijcai.2022/764. URL https://doi.org/10.24963/ijcai.2022/764.

Ilaria Ferrari, Giacomo Frisoni, Paolo Italiani, Gianluca Moro, and Claudio Sartori. Comprehensive analysis of knowledge graph embedding techniques benchmarked on link prediction. Electronics, 11(23), 2022. ISSN 2079-9292. doi: 10.3390/electronics11233866. URL https://www.mdpi.com/2079-9292/11/23/3866.

Joshua C Fjelstul. The evolution of european union law: A new data set on the acquis communautaire. European Union Politics, 20(4):670–691, 2019.

Luciano Floridi and Massimo Chiriatti. GPT-3: its nature, scope, limits, and consequences. Minds Mach., 30(4):681–694, 2020. doi: 10.1007/S11023-020-09548-1. URL https://doi.org/10.1007/s11023-020-09548-1.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. Match your words! A study of lexical matching in neural information retrieval. In Matthias Hagen, Suzan Verberne, Craig Macdonald, Christin Seifert, Krisztian Balog, Kjetil Nørvåg, and Vinay Setty, editors, Advances in Information Retrieval - 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10-14, 2022, Proceedings, Part II, volume 13186 of Lecture Notes in Computer Science, pages 120–127. Springer, 2022. doi: 10.1007/978-3-030-99739-7_14. URL https://doi.org/10.1007/978-3-030-99739-7_14.

Avital Friedland, Jonathan Zeltser, and Omer Levy. Are mutually intelligible languages easier to translate? CoRR, abs/2201.13072, 2022. URL https://arxiv.org/abs/2201.13072.

Giacomo Frisoni and Gianluca Moro. Phenomena explanation from text: Unsupervised learning of interpretable and statistically significant knowledge. In Slimane Hammoudi, Christoph Quix, and Jorge Bernardino, editors, Data Management Technologies and Applications - 9th International Conference, DATA 2020, Virtual Event, July 7-9, 2020, Revised Selected Papers, volume 1446 of Communications in Computer and Information Science, pages 293–318. Springer, 2020. doi: 10.1007/978-3-030-83014-4_14. URL https://doi.org/10.1007/978-3-030-83014-4_14.

Giacomo Frisoni, Gianluca Moro, and Antonella Carbonaro. Learning interpretable and statistically significant knowledge from unlabeled corpora of social text messages: A novel methodology of descriptive text mining. In Slimane Hammoudi, Christoph Quix, and Jorge Bernardino, editors, Proceedings of the 9th International Conference on Data Science, Technology and Applications, DATA 2020, Lieusaint, Paris, France, July 7-9, 2020, pages 121–132. SciTePress, 2020a. doi: 10.5220/0009892001210132. URL https://doi.org/10.5220/0009892001210132.

Giacomo Frisoni, Gianluca Moro, and Antonella Carbonaro. Unsupervised descriptive text mining for knowledge graph learning. In Ana L. N. Fred and Joaquim Filipe, editors, Proceedings of the 12th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K 2020, Volume 1: KDIR, Budapest, Hungary, November 2-4, 2020, pages 316–324. SCITEPRESS, 2020b. doi: 10.5220/0010153603160324. URL https://doi.org/10.5220/0010153603160324.

Giacomo Frisoni, Gianluca Moro, and Antonella Carbonaro. A survey on event extraction for natural language understanding: Riding the biomedical literature wave. IEEE Access, 9:160721–160757, 2021. doi: 10.1109/ACCESS.2021.3130956. URL https://doi.org/10.1109/ACCESS.2021.3130956.

Giacomo Frisoni, Antonella Carbonaro, Gianluca Moro, Andrea Zammarchi, and Marco Avagnano. Nlgmetricverse: An end-to-end library for evaluating natural language generation. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022, pages 3465–3479. International Committee on Computational Linguistics, 2022a. URL https://aclanthology.org/2022.coling-1.306.

Giacomo Frisoni, Paolo Italiani, Francesco Boschi, and Gianluca Moro. Enhancing biomedical scientific reviews summarization with graph-based factual evidence extracted from papers. In Alfredo Cuzzocrea, Oleg Gusikhin, Wil M. P. van der Aalst, and Slimane Hammoudi, editors, Proceedings of the 11th International Conference on Data Science, Technology and Applications, DATA 2022, Lisbon, Portugal, July 11-13, 2022, pages 168–179. SCITEPRESS, 2022b. doi: 10.5220/0011354900003269. URL https://doi.org/10.5220/0011354900003269.

Giacomo Frisoni, Miki Mizutani, Gianluca Moro, and Lorenzo Valgimigli. Bioreader: a retrieval-enhanced text-to-text transformer for biomedical literature. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, Proceedings of the 2022 Conference on Empirical Methods in Natural Language

Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, pages 5770–5793. Association for Computational Linguistics, 2022c. doi: 10.18653/V1/2022.EMNLP-MAIN.390. URL https://doi.org/10.18653/v1/2022.emnlp-main.390.

Giacomo Frisoni, Gianluca Moro, and Lorenzo Balzani. Text-to-text extraction and verbalization of biomedical event graphs. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022, pages 2692–2710. International Committee on Computational Linguistics, 2022d. URL https://aclanthology.org/2022.coling-1.238.

Giacomo Frisoni, Gianluca Moro, Giulio Carlassare, and Antonella Carbonaro. Unsupervised event graph representation and similarity learning on biomedical literature. Sensors, 22(1):3, 2022e. doi: 10.3390/S22010003. URL https://doi.org/10.3390/s22010003.

Giacomo Frisoni, Paolo Italiani, Stefano Salvatori, and Gianluca Moro. Cogito ergo summ: Abstractive summarization of biomedical papers via semantic parsing graphs and consistency rewards. In Brian Williams, Yiling Chen, and Jennifer Neville, editors, Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023, pages 12781–12789. AAAI Press, 2023. doi: 10.1609/AAAI.V37I11.26503. URL https://doi.org/10.1609/aaai.v37i11.26503.

Giacomo Frisoni, Luca Ragazzi, David Cohen, Gianluca Moro, Antonella Carbonaro, and Claudio Sartori. Abstractive summarization through the Prism of decoding strategies, 2024. Submitted to Computational Linguistics.

Andrea Galassi, Kasper Drazewski, Marco Lippi, and Paolo Torroni. Cross-lingual annotation projection in legal texts. In Donia Scott, Núria Bel, and Chengqing Zong, editors, Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020, pages 915–926. International Committee on Computational Linguistics, 2020. doi: 10.18653/V1/2020.COLING-MAIN.79. URL https://doi.org/10.18653/v1/2020.coling-main.79.

Filippo Galgani and Achim G. Hoffmann. LEXA: towards automatic legal citation classification. In Jiuyong Li, editor, AI 2010: Advances in Artificial Intelligence - 23rd Australasian Joint Conference, Adelaide, Australia, December 7-10, 2010. Proceedings, volume 6464 of Lecture Notes in Computer Science, pages 445–454. Springer, 2010. doi: 10.1007/978-3-642-17432-2_45. URL https://doi.org/10.1007/978-3-642-17432-2_45.

Filippo Galgani, Paul Compton, and Achim Hoffmann. Combining different summarization techniques for legal text. In Proceedings of the workshop on innovative hybrid approaches to the processing of textual data, pages 115–123. Association for Computational Linguistics, 2012a.

Filippo Galgani, Paul Compton, and Achim G. Hoffmann. Towards automatic generation of catchphrases for legal case reports. In Alexander F. Gelbukh, editor, Computational Linguistics and Intelligent Text Processing - 13th International Conference, CICLing 2012, New Delhi, India, March 11-17, 2012, Proceedings, Part II, volume 7182 of Lecture Notes in Computer Science, pages 414–425. Springer, 2012b. doi: 10.1007/978-3-642-28601-8_35. URL https://doi.org/10.1007/978-3-642-28601-8_35.

Filippo Galgani, Paul Compton, and Achim G. Hoffmann. Knowledge acquisition for categorization of legal case reports. In Deborah Richards and Byeong Ho Kang, editors, Knowledge Management and

Acquisition for Intelligent Systems - 12th Pacific Rim Knowledge Acquisition Workshop, PKAW 2012, Kuching, Malaysia, September 5-6, 2012. Proceedings, volume 7457 of Lecture Notes in Computer Science, pages 118–132. Springer, 2012c. doi: 10.1007/978-3-642-32541-0_10. URL https://doi.org/10.1007/978-3-642-32541-0_10.

Filippo Galgani, Paul Compton, and Achim G. Hoffmann. Citation based summarisation of legal texts. In Patricia Anthony, Mitsuru Ishizuka, and Dickson Lukose, editors, PRICAI 2012: Trends in Artificial Intelligence - 12th Pacific Rim International Conference on Artificial Intelligence, Kuching, Malaysia, September 3-7, 2012. Proceedings, volume 7458 of Lecture Notes in Computer Science, pages 40–52. Springer, 2012d. doi: 10.1007/978-3-642-32695-0_6. URL https://doi.org/10.1007/978-3-642-32695-0_6.

Filippo Galgani, Paul Compton, and Achim G. Hoffmann. Summarization based on bi-directional citation analysis. Inf. Process. Manag., 51(1):1–24, 2015. doi: 10.1016/J.IPM.2014.08.001. URL https://doi.org/10.1016/j.ipm.2014.08.001.

Federico Galli, Giulia Grundler, Alessia Fidelangeli, Andrea Galassi, Francesca Lagioia, Elena Palmieri, Federico Ruggeri, Giovanni Sartor, and Paolo Torroni. Predicting outcomes of italian VAT decisions. In Enrico Francesconi, Georg Borges, and Christoph Sorge, editors, Legal Knowledge and Information Systems - JURIX 2022: The Thirty-fifth Annual Conference, Saarbrücken, Germany, 14-16 December 2022, volume 362 of Frontiers in Artificial Intelligence and Applications, pages 188–193. IOS Press, 2022. doi: 10.3233/FAIA220465. URL https://doi.org/10.3233/FAIA220465.

Jiahui Gao, Renjie Pi, Yong Lin, Hang Xu, Jiacheng Ye, Zhiyong Wu, Weizhong Zhang, Xiaodan Liang, Zhenguo Li, and Lingpeng Kong. Self-guided noise-free data generation for efficient zero-shot learning. In The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net, 2023a. URL https://openreview.net/pdf?id=h5OpjGd_lo6.

Yingbo Gao, Weiyue Wang, Christian Herold, Zijian Yang, and Hermann Ney. Towards a better understanding of label smoothing in neural machine translation. In Kam-Fai Wong, Kevin Knight, and Hua Wu, editors, Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2020, Suzhou, China, December 4-7, 2020, pages 212–223. Association for Computational Linguistics, 2020. URL https://aclanthology.org/2020.aacl-main.25/.

Yuan Gao, Ruili Wang, and Feng Hou. How to design translation prompts for chatgpt: An empirical study. CoRR, abs/2304.02182, 2023b. doi: 10.48550/ARXIV.2304.02182. URL https://doi.org/10.48550/arXiv.2304.02182.

Sebastian Gehrmann, Yuntian Deng, and Alexander M. Rush. Bottom-up abstractive summarization. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, pages 4098–4109. Association for Computational Linguistics, 2018. doi: 10.18653/V1/D18-1443. URL https://doi.org/10.18653/v1/d18-1443.

Iacopo Ghinassi, Lin Wang, Chris Newell, and Matthew Purver. Lessons learnt from linear text segmentation: a fair comparison of architectural and sentence encoding strategies for successful segmentation. In Ruslan Mitkov and Galia Angelova, editors, Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing, RANLP 2023, Varna, Bulgaria, 4-6 September 2023, pages 408–418. INCOMA Ltd., Shoumen, Bulgaria, 2023. URL https://aclanthology.org/2023.ranlp-1.46.

Alexios Gidiotis and Grigorios Tsoumakas. A divide-and-conquer approach to the summarization of long documents. IEEE ACM Trans. Audio Speech Lang. Process., 28:3029–3040, 2020. doi: 10.1109/TASLP.2020.3037401. URL https://doi.org/10.1109/TASLP.2020.3037401.

Jorge Martínez Gil. A survey on legal question-answering systems. Comput. Sci. Rev., 48:100552, 2023. doi: 10.1016/J.COSREV.2023.100552. URL https://doi.org/10.1016/j.cosrev.2023.100552.

Mario Giulianelli, Joris Baan, Wilker Aziz, Raquel Fernández, and Barbara Plank. What comes next? evaluating uncertainty in neural text generators against human production variability. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023, pages 14349–14371. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.emnlp-main.887.

Tomas Goldsack, Zhihao Zhang, Chenghua Lin, and Carolina Scarton. Making science simple: Corpora for the lay summarisation of scientific literature. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, pages 10589–10604. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.724. URL https://doi.org/10.18653/v1/2022.emnlp-main.724.

Ning Gong and Nianmin Yao. A generalized decoding method for neural text generation. Comput. Speech Lang., 81:101503, 2023. doi: 10.1016/J.CSL.2023.101503. URL https://doi.org/10.1016/j.csl.2023.101503.

José Ángel González, Annie Louis, and Jackie Chi Kit Cheung. Source-summary entity aggregation in abstractive summarization. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022, pages 6019–6034. International Committee on Computational Linguistics, 2022. URL https://aclanthology.org/2022.coling-1.526.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. Int. J. Comput. Vis., 129(6):1789–1819, 2021. doi: 10.1007/S11263-021-01453-Z. URL https://doi.org/10.1007/s11263-021-01453-z.

Guido Governatori, Trevor J. M. Bench-Capon, Bart Verheij, Michal Araszkiewicz, Enrico Francesconi, and Matthias Grabmair. Thirty years of artificial intelligence and law: the first decade. Artif. Intell. Law, 30(4):481–519, 2022. doi: 10.1007/S10506-022-09329-4. URL https://doi.org/10.1007/s10506-022-09329-4.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. News summarization and evaluation in the era of GPT-3. CoRR, abs/2209.12356, 2022a. doi: 10.48550/ARXIV.2209.12356. URL https://doi.org/10.48550/arXiv.2209.12356.

Tanya Goyal, Nazneen Rajani, Wenhao Liu, and Wojciech Kryscinski. Hydrasum: Disentangling style features in text summarization with multi-decoder models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, pages 464–479. Association for Computational Linguistics, 2022b. doi: 10.18653/V1/2022.EMNLP-MAIN.30. URL https://doi.org/10.18653/v1/2022.emnlp-main.30.

Quentin Grail, Julien Perez, and Éric Gaussier. Globalizing bert-based transformer architectures for long document summarization. In Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty, editors, Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021, pages 1792–1810. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EACL-MAIN.154. URL https://doi.org/10.18653/v1/2021.eacl-main.154.

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. CoRR, abs/1410.5401, 2014. URL http://arxiv.org/abs/1410.5401.

Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwinska, Sergio Gomez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John P. Agapiou, Adrià Puigdomènech Badia, Karl Moritz Hermann, Yori Zwols, Georg Ostrovski, Adam Cain, Helen King, Christopher Summerfield, Phil Blunsom, Koray Kavukcuoglu, and Demis Hassabis. Hybrid computing using a neural network with dynamic external memory. Nat., 538(7626):471–476, 2016. doi: 10.1038/NATURE20101. URL https://doi.org/10.1038/nature20101.

Maarten Grootendorst. Keybert: Minimal keyword extraction with bert., 2020. URL https://doi.org/10.5281/zenodo.4461265.

Claire Grover, Ben Hachey, and Ian Hughson. The HOLJ corpus. supporting summarisation of legal texts. In COLING, pages 47–54, Geneva, Switzerland, aug 29 2004. COLING. URL https://aclanthology.org/W04-1907.

Max Grusky, Mor Naaman, and Yoav Artzi. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers), pages 708–719. Association for Computational Linguistics, 2018. doi: 10.18653/V1/N18-1065. URL https://doi.org/10.18653/v1/n18-1065.

Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, and Tsuhan Chen. Recent advances in convolutional neural networks. Pattern Recognit., 77:354–377, 2018. doi: 10.1016/J.PATCOG.2017.10.013. URL https://doi.org/10.1016/j.patcog.2017.10.013.

Nianlong Gu, Elliott Ash, and Richard H. R. Hahnloser. Memsum: Extractive summarization of long documents using multi-step episodic markov decision processes. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 6507–6522. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.ACL-LONG.450. URL https://doi.org/10.18653/v1/2022.acl-long.450.

Xiaotao Gu, Yuning Mao, Jiawei Han, Jialu Liu, You Wu, Cong Yu, Daniel Finnie, Hongkun Yu, Jiaqi Zhai, and Nicholas Zukoski. Generating representative headlines for news stories. In Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen, editors, WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020, pages 1773–1784. ACM / IW3C2, 2020. doi: 10.1145/3366423.3380247. URL https://doi.org/10.1145/3366423.3380247.

Xin Guan, Biwei Cao, Qingqing Gao, Zheng Yin, Bo Liu, and Jiuxin Cao. CORN: co-reasoning network for commonsense question answering. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022, pages 1677–1686. International Committee on Computational Linguistics, 2022. URL https://aclanthology.org/2022.coling-1.144.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. The false promise of imitating proprietary llms. CoRR, abs/2305.15717, 2023. doi: 10.48550/ARXIV.2305.15717. URL https://doi.org/10.48550/arXiv.2305.15717.

Neel Guha, Daniel E. Ho, Julian Nyarko, and Christopher Ré. Legalbench: Prototyping a collaborative benchmark for legal reasoning. CoRR, abs/2209.06120, 2022. doi: 10.48550/ARXIV.2209.06120. URL https://doi.org/10.48550/arXiv.2209.06120.

Çaglar Gülçehre, Sarath Chandar, Kyunghyun Cho, and Yoshua Bengio. Dynamic neural turing machine with continuous and discrete addressing schemes. Neural Comput., 30(4), 2018. doi: 10.1162/NECO_A_01060. URL https://doi.org/10.1162/neco_a_01060.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. Soft layer-specific multi-task summarization with entailment and question generation. In Iryna Gurevych and Yusuke Miyao, editors, Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers, pages 687–697. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1064. URL https://www.aclweb.org/anthology/P18-1064/.

Mandy Guo, Joshua Ainslie, David C. Uthus, Santiago Ontañón, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. Longt5: Efficient text-to-text transformer for long sequences. In Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz, editors, Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022, pages 724–736. Association for Computational Linguistics, 2022a. doi: 10.18653/V1/2022.FINDINGS-NAACL.55. URL https://doi.org/10.18653/v1/2022.findings-naacl.55.

Zhen Guo, Peiqi Wang, Yanwei Wang, and Shangdi Yu. Dr. llama: Improving small language models on pubmedqa via generative data augmentation. CoRR, abs/2305.07804, 2023. doi: 10.48550/ARXIV.2305.07804. URL https://doi.org/10.48550/arXiv.2305.07804.

Zhijiang Guo, Michael Sejr Schlichtkrull, and Andreas Vlachos. A survey on automated fact-checking. Trans. Assoc. Comput. Linguistics, 10:178–206, 2022b. doi: 10.1162/tacl_a_00454. URL https://doi.org/10.1162/tacl_a_00454.

Udit Gupta, Young Geun Kim, Sylvia Lee, Jordan Tse, Hsien-Hsin S. Lee, Gu-Yeon Wei, David Brooks, and Carole-Jean Wu. Chasing carbon: The elusive environmental footprint of computing. IEEE Micro, 42(4):37–47, 2022. doi: 10.1109/MM.2022.3163226. URL https://doi.org/10.1109/MM.2022.3163226.

Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pages 8342–8360. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.ACL-MAIN.740. URL https://doi.org/10.18653/v1/2020.acl-main.740.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. REALM: retrieval-augmented language model pre-training. CoRR, abs/2002.08909, 2020. URL https://arxiv.org/abs/2002.08909.

Raluca-Maria Hampau, Maurits Kaptein, Robin van Emden, Thomas Rost, and Ivano Malavolta. An empirical study on the performance and energy consumption of AI containerization strategies for computer-vision tasks on the edge. In Miroslaw Staron, Christian Berger, Jocelyn Simmonds, and Rafael Prikladnicki, editors, EASE 2022: The International Conference on Evaluation and Assessment in Software Engineering 2022, Gothenburg, Sweden, June 13 - 15, 2022, pages 50–59. ACM, 2022. doi: 10.1145/3530019.3530025. URL https://doi.org/10.1145/3530019.3530025.

Xu-Wang Han, Hai-Tao Zheng, Jin-Yuan Chen, and Cong-Zhi Zhao. Diverse decoding for abstractive document summarization. Applied Sciences, 9(3), 2019. ISSN 2076-3417. doi: 10.3390/app9030386. URL https://www.mdpi.com/2076-3417/9/3/386.

Chuzhan Hao, Minghui Xie, and Peng Zhang. Acenet: Attention guided commonsense reasoning on hybrid knowledge graph. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, pages 8461–8471. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.579. URL https://doi.org/10.18653/v1/2022.emnlp-main.579.

Pengcheng He, Baolin Peng, Song Wang, Yang Liu, Ruochen Xu, Hany Hassan, Yu Shi, Chenguang Zhu, Wayne Xiong, Michael Zeng, Jianfeng Gao, and Xuedong Huang. Z-code++: A pre-trained language model optimized for abstractive summarization. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023, pages 5095–5112. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.279. URL https://doi.org/10.18653/v1/2023.acl-long.279.

Michael A. Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. A survey on recent approaches for natural language processing in low-resource scenarios. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021, pages 2545–2568. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.NAACL-MAIN.201. URL https://doi.org/10.18653/v1/2021.naacl-main.201.

Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. Towards the systematic reporting of the energy and carbon footprints of machine learning. CoRR, abs/2002.05651, 2020. URL https://arxiv.org/abs/2002.05651.

Peter Henderson, Mark S. Krass, Lucia Zheng, Neel Guha, Christopher D. Manning, Dan Jurafsky, and Daniel E. Ho. Pile of law: Learning responsible data filtering from the law and a 256gb open-source legal dataset. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/bc218a0c656e49d4b086975a9c785f47-Abstract-Datasets_and_Benchmarks.html.

John Hewitt, Christopher D. Manning, and Percy Liang. Truncation sampling as language model desmoothing. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, pages 3414–3427. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.FINDINGS-EMNLP.249. URL https://doi.org/10.18653/v1/2022.findings-emnlp.249.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. CoRR, abs/1503.02531, 2015. URL http://arxiv.org/abs/1503.02531.

Thomas Hofmann. Probabilistic latent semantic analysis. In Kathryn B. Laskey and Henri Prade, editors, UAI '99: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, Stockholm, Sweden, July 30 - August 1, 1999, pages 289–296. Morgan Kaufmann, 1999. URL https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=179&proceeding_id=15.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge graphs. ACM Comput. Surv., 54(4), jul 2021. ISSN 0360-0300. doi: 10.1145/3447772. URL https://doi.org/10.1145/3447772.

Chris Hokamp, Demian Gholipour Ghalandari, Nghia The Pham, and John Glover. Dyne: Dynamic ensemble decoding for multi-document summarization. CoRR, abs/2006.08748, 2020. URL https://arxiv.org/abs/2006.08748.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

Nils Holzenberger, Andrew Blair-Stanek, and Benjamin Van Durme. A dataset for statutory reasoning in tax law entailment and question answering. In Nikolaos Aletras, Ion Androutsopoulos, Leslie Barrett, Adam Meyers, and Daniel Preotiuc-Pietro, editors, Proceedings of the Natural Legal Language Processing Workshop 2020 co-located with the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD 2020), Virtual Workshop, August 24, 2020, volume 2645 of CEUR Workshop Proceedings, pages 31–38. CEUR-WS.org, 2020. URL https://ceur-ws.org/Vol-2645/paper5.pdf.

Linmei Hu, Zeyi Liu, Ziwang Zhao, Lei Hou, Liqiang Nie, and Juanzi Li. A survey of knowledge-enhanced pre-trained language models. CoRR, abs/2211.05994, 2022a. doi: 10.48550/ARXIV.2211.05994. URL https://doi.org/10.48550/arXiv.2211.05994.

Ziniu Hu, Yichong Xu, Wenhao Yu, Shuohang Wang, Ziyi Yang, Chenguang Zhu, Kai-Wei Chang, and Yizhou Sun. Empowering language models with knowledge graph reasoning for open-domain question answering. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, pages 9562–9581. Association for Computational Linguistics, 2022b. doi: 10.18653/V1/2022.EMNLP-MAIN.650. URL https://doi.org/10.18653/v1/2022.emnlp-main.650.

Kung-Hsiang Huang, Siffi Singh, Xiaofei Ma, Wei Xiao, Feng Nan, Nicholas Dingwall, William Yang Wang, and Kathleen R. McKeown. SWING: balancing coverage and faithfulness for dialogue summarization. In Andreas Vlachos and Isabelle Augenstein, editors, Findings of the Association for Computational Linguistics: EACL 2023, Dubrovnik, Croatia, May 2-6, 2023, pages 512–525. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-EACL.37. URL https://doi.org/10.18653/v1/2023.findings-eacl.37.

Luyang Huang, Shuyang Cao, Nikolaus Nova Parulian, Heng Ji, and Lu Wang. Efficient attentions for long document summarization. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021, pages 1419–1436. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.NAACL-MAIN.112. URL https://doi.org/10.18653/v1/2021.naacl-main.112.

Yuxin Huang, Zhengtao Yu, Junjun Guo, Zhiqiang Yu, and Yantuan Xian. Legal public opinion news abstractive summarization by incorporating topic information. Int. J. Mach. Learn. Cybern., 11(9):2039–2050, 2020. doi: 10.1007/S13042-020-01093-8. URL https://doi.org/10.1007/s13042-020-01093-8.

Taehun Huh and Youngjoong Ko. Efficient framework for low-resource abstractive summarization by meta-transfer learning and pointer-generator networks. Expert Syst. Appl., 234:121029, 2023. doi: 10.1016/J.ESWA.2023.121029. URL https://doi.org/10.1016/j.eswa.2023.121029.

Wonseok Hwang, Dongjun Lee, Kyoungyeon Cho, Hanuhl Lee, and Minjoon Seo. A multi-task benchmark for korean legal language understanding and judgement prediction. In

Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/d15abd14d5894eebd185b756541d420e-Abstract-Datasets_and_Benchmarks.html.

Oana Ignat, Zhijing Jin, Artem Abzaliev, Laura Biester, Santiago Castro, Naihao Deng, Xinyi Gao, Aylin Gunal, Jacky He, Ashkan Kazemi, Muhammad Khalifa, Namho Koh, Andrew Lee, Siyang Liu, Do June Min, Shinka Mori, Joan Nwatu, Verónica Pérez-Rosas, Siqi Shen, Zekun Wang, Winston Wu, and Rada Mihalcea. A phd student's perspective on research in NLP in the era of very large language models. CoRR, abs/2305.12544, 2023. doi: 10.48550/ARXIV.2305.12544. URL https://doi.org/10.48550/arXiv.2305.12544.

Uswa Ihsan, Humaira Ashraf, and NZ Jhanjhi. Survey on multi-document summarization: Systematic literature review. CoRR, abs/2312.12915, 2023. doi: 10.48550/ARXIV.2312.12915. URL https://doi.org/10.48550/arXiv.2312.12915.

Daphne Ippolito, Reno Kriz, João Sedoc, Maria Kustikova, and Chris Callison-Burch. Comparison of diverse decoding methods from conditional language models. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pages 3752–3762. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1365. URL https://doi.org/10.18653/v1/p19-1365.

Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerdà, Javier Jorge, Nahuel Roselló, Adrià Giménez, Albert Sanchís, Jorge Civera, and Alfons Juan. Europarl-st: A multilingual corpus for speech translation of parliamentary debates. In 2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020, pages 8229–8233. IEEE, 2020. doi: 10.1109/ICASSP40776.2020.9054626. URL https://doi.org/10.1109/ICASSP40776.2020.9054626.

Paolo Italiani, Luca Ragazzi, and Gianluca Moro. Enhancing legal question answering with data generation and knowledge distillation from large language models, 2024. Submitted to Artificial Intelligence and Law.

Maor Ivgi, Uri Shaham, and Jonathan Berant. Efficient long-text understanding with short-text models. CoRR, abs/2208.00748, 2022. doi: 10.48550/ARXIV.2208.00748. URL https://doi.org/10.48550/arXiv.2208.00748.

Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty, editors, Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021, pages 874–880. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EACL-MAIN.74. URL https://doi.org/10.18653/v1/2021.eacl-main.74.

Deepali Jain, Malaya Dutta Borah, and Anupam Biswas. Automatic summarization of legal bills: A comparative analysis of classical extractive approaches. In 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), pages 394–400, 2021a. doi: 10.1109/ICCCIS51004.2021.9397119.

Deepali Jain, Malaya Dutta Borah, and Anupam Biswas. Summarization of legal documents: Where are we now and the way forward. Comput. Sci. Rev., 40:100388, 2021b. doi: 10.1016/J.COSREV.2021.100388. URL https://doi.org/10.1016/j.cosrev.2021.100388.

Raghav Jain, Anubhav Jangra, Sriparna Saha, and Adam Jatowt. A survey on medical document summarization. CoRR, abs/2212.01669, 2022. doi: 10.48550/ARXIV.2212.01669. URL https://doi.org/10.48550/arXiv.2212.01669.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net, 2017. URL https://openreview.net/forum?id=rkE3y85ee.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. ACM Trans. Inf. Syst., 20(4):422–446, 2002. doi: 10.1145/582415.582418. URL http://doi.acm.org/10.1145/582415.582418.

Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. Perplexity—a measure of the difficulty of speech recognition tasks. The Journal of the Acoustical Society of America, 62(S1):S63–S63, 1977.

Xin Ji and Wen Zhao. SKGSUM: abstractive document summarization with semantic knowledge graphs. In International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021, pages 1–8. IEEE, 2021. doi: 10.1109/IJCNN52387.2021.9533494. URL https://doi.org/10.1109/IJCNN52387.2021.9533494.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. ACM Comput. Surv., 55(12):248:1–248:38, 2023. doi: 10.1145/3571730. URL https://doi.org/10.1145/3571730.

Ruipeng Jia, Yanan Cao, Hengzhu Tang, Fang Fang, Cong Cao, and Shi Wang. Neural extractive summarization with hierarchical attentive heterogeneous graph network. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pages 3622–3631. Association for Computational Linguistics, 2020a. doi: 10.18653/V1/2020.EMNLP-MAIN.295. URL https://doi.org/10.18653/v1/2020.emnlp-main.295.

Zhihao Jia, Sina Lin, Mingyu Gao, Matei Zaharia, and Alex Aiken. Improving the accuracy, scalability, and performance of graph neural networks with roc. In Inderjit S. Dhillon, Dimitris S. Papailiopoulos, and Vivienne Sze, editors, Proceedings of Machine Learning and Systems 2020, MLSys 2020, Austin, TX, USA, March 2-4, 2020. mlsys.org, 2020b. URL https://proceedings.mlsys.org/book/300.pdf.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. CoRR, abs/2310.06825, 2023. doi: 10.48550/ARXIV.2310.06825. URL https://doi.org/10.48550/arXiv.2310.06825.

Jinhao Jiang, Kun Zhou, Ji-Rong Wen, and Wayne Xin Zhao. $great truths are always simple: $ A rather simple knowledge encoder for enhancing the commonsense reasoning capacity of pre-trained models. In Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz, editors, Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022, pages 1730–1741. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.FINDINGS-NAACL.131. URL https://doi.org/10.18653/v1/2022.findings-naacl.131.

Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, and Zhaopeng Tu. Is chatgpt A good translator? A preliminary study. CoRR, abs/2301.08745, 2023. doi: 10.48550/ARXIV.2301.08745. URL https://doi.org/10.48550/arXiv.2301.08745.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling BERT for natural language understanding. In Trevor Cohn, Yulan He, and Yang Liu, editors, Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020, volume EMNLP 2020 of Findings of ACL, pages 4163–4174. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.FINDINGS-EMNLP.372. URL https://doi.org/10.18653/v1/2020.findings-emnlp.372.

Hanqi Jin, Tianming Wang, and Xiaojun Wan. Multi-granularity interaction network for extractive and abstractive multi-document summarization. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pages 6244–6254. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.ACL-MAIN.556. URL https://doi.org/10.18653/v1/2020.acl-main.556.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. IEEE Trans. Big Data, 7(3):535–547, 2021. doi: 10.1109/TBDATA.2019.2921572. URL https://doi.org/10.1109/TBDATA.2019.2921572.

Katikapalli Subramanyam Kalyan, Ajit Rajasekharan, and Sivanesan Sangeetha. AMMUS : A survey of transformer-based pretrained models in natural language processing. CoRR, abs/2108.05542, 2021. URL https://arxiv.org/abs/2108.05542.

Ambedkar Kanapala, Sukomal Pal, and Rajendra Pamula. Text summarization from legal documents: a survey. Artif. Intell. Rev., 51(3):371–402, 2019. doi: 10.1007/S10462-017-9566-2. URL https://doi.org/10.1007/s10462-017-9566-2.

Yoshinobu Kano, Mi-Young Kim, Masaharu Yoshioka, Yao Lu, Juliano Rabelo, Naoki Kiyota, Randy Goebel, and Ken Satoh. COLIEE-2018: evaluation of the competition on legal information extraction and entailment. In Kazuhiro Kojima, Maki Sakamoto, Koji Mineshima, and Ken Satoh, editors, New Frontiers in Artificial Intelligence - JSAI-isAI 2018 Workshops, JURISIN, AI-Biz, SKL, LENLS, IDAA, Yokohama, Japan, November 12-14, 2018, Revised Selected Papers, volume 11717 of Lecture Notes in Computer Science, pages 177–192. Springer, 2018. doi: 10.1007/978-3-030-31605-1_14. URL https://doi.org/10.1007/978-3-030-31605-1_14.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pages 6769–6781. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.EMNLP-MAIN.550. URL https://doi.org/10.18653/v1/2020.emnlp-main.550.

Daniel Martin Katz, Dirk Hartung, Lauritz Gerlach, Abhik Jana, and Michael J. Bommarito II. Natural language processing in the legal domain. CoRR, abs/2302.12039, 2023. doi: 10.48550/ARXIV.2302.12039. URL https://doi.org/10.48550/arXiv.2302.12039.

S Kayalvizhi and D Thenmozhi. Deep learning approach for extracting catch phrases from legal documents. In Neural Networks for Natural Language Processing, pages 143–158. IGI Global, 2020.

Bilal Khan, Zohaib Ali Shah, Muhammad Usman, Inayat Khan, and Badam Niazi. Exploring the landscape of automatic text summarization: A comprehensive survey. IEEE Access, 11:109819–109840, 2023. doi: 10.1109/ACCESS.2023.3322188. URL https://doi.org/10.1109/ACCESS.2023.3322188.

Khalid S Khan, Regina Kunz, Jos Kleijnen, and Gerd Antes. Five steps to conducting a systematic review. Journal of the royal society of medicine, 96(3):118–121, 2003.

Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. Reframing instructional prompts to gptk's language. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 589–612. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.FINDINGS-ACL.50. URL https://doi.org/10.18653/v1/2022.findings-acl.50.

Dahyun Kim, Chanjun Park, Sanghoon Kim, Wonsung Lee, Wonho Song, Yunsu Kim, Hyeonwoo Kim, Yungi Kim, Hyeonju Lee, Jihoo Kim, Changbae Ahn, Seonghoon Yang, Sukyung Lee, Hyunbyung Park, Gyoungjin Gim, Mikyoung Cha, Hwalsuk Lee, and Sunghun Kim. SOLAR 10.7b: Scaling large language models with simple yet effective depth up-scaling. CoRR, abs/2312.15166, 2023. doi: 10.48550/ARXIV.2312.15166. URL https://doi.org/10.48550/arXiv.2312.15166.

Daniel King, Zejiang Shen, Nishant Subramani, Daniel S. Weld, Iz Beltagy, and Doug Downey. Don't say what you don't know: Improving the consistency of abstractive summarization by constraining beam search. CoRR, abs/2203.08436, 2022. doi: 10.48550/ARXIV.2203.08436. URL https://doi.org/10.48550/arXiv.2203.08436.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020. URL https://openreview.net/forum?id=rkgNKkHtvB.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. Opennmt: Open-source toolkit for neural machine translation. In Mohit Bansal and Heng Ji, editors, Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, System Demonstrations, pages 67–72. Association for Computational Linguistics, 2017. doi: 10.18653/V1/P17-4012. URL https://doi.org/10.18653/v1/P17-4012.

Tshepho Koboyatshwene, Moemedi Lefoane, and Lakshmi Narasimhan. Machine learning approaches for catchphrase extraction in legal documents. In Prasenjit Majumder, Mandar Mitra, Parth Mehta, and Jainisha Sankhavara, editors, Working notes of FIRE 2017 - Forum for Information Retrieval Evaluation, Bangalore, India, December 8-10, 2017, volume 2036 of CEUR Workshop Proceedings, pages 95–98. CEUR-WS.org, 2017. URL http://ceur-ws.org/Vol-2036/T3-11.pdf.

Bernard Koch, Emily Denton, Alex Hanna, and Jacob G. Foster. Reduced, reused and recycled: The life of a dataset in machine learning research. In Joaquin Vanschoren and Sai-Kit Yeung, editors, Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/3b8a614226a953a8cd9526fca6fe9ba5-Abstract-round2.html.

Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In Proceedings of Machine Translation Summit X: Papers, MTSummit 2005, Phuket, Thailand, September 13-15, 2005, pages 79–86, 2005. URL https://aclanthology.org/2005.mtsummit-papers.11.

Huan Yee Koh, Jiaxin Ju, He Zhang, Ming Liu, and Shirui Pan. How far are we from robust long abstractive summarization? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, pages 2682–2698. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.172. URL https://doi.org/10.18653/v1/2022.emnlp-main.172.

Huan Yee Koh, Jiaxin Ju, Ming Liu, and Shirui Pan. An empirical survey on long document summarization: Datasets, models, and metrics. ACM Comput. Surv., 55(8):154:1–154:35, 2023a. doi: 10.1145/3545176. URL https://doi.org/10.1145/3545176.

Jing Yu Koh, Daniel Fried, and Ruslan Salakhutdinov. Generating images with multimodal language models. CoRR, abs/2305.17216, 2023b. doi: 10.48550/ARXIV.2305.17216. URL https://doi.org/10.48550/arXiv.2305.17216.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/8bb0d291acd4acf06ef112099c16f326-Abstract-Conference.html.

Anastassia Kornilova and Vlad Eidelman. Billsum: A corpus for automatic summarization of US legislation. CoRR, abs/1910.00523, 2019. URL http://arxiv.org/abs/1910.00523.

Fajri Koto, Timothy Baldwin, and Jey Han Lau. Lipkey: A large-scale news dataset for absent keyphrases generation and abstractive summarization. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022, pages 3427–3437. International Committee on Computational Linguistics, 2022. URL https://aclanthology.org/2022.coling-1.303.

Wojciech Kryscinski, Romain Paulus, Caiming Xiong, and Richard Socher. Improving abstraction in text summarization. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, pages 1808–1817. Association for Computational Linguistics, 2018. doi: 10.18653/v1/d18-1207. URL https://doi.org/10.18653/v1/d18-1207.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. Evaluating the factual consistency of abstractive text summarization. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pages 9332–9346. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.EMNLP-MAIN.750. URL https://doi.org/10.18653/v1/2020.emnlp-main.750.

Wojciech Kryscinski, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. BOOKSUM: A collection of datasets for long-form narrative summarization. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, pages 6536–6558. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.FINDINGS-EMNLP.488. URL https://doi.org/10.18653/v1/2022.findings-emnlp.488.

Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Eduardo Blanco and Wei Lu, editors, Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018, pages 66–71. Association for Computational Linguistics, 2018. doi: 10.18653/V1/D18-2012. URL https://doi.org/10.18653/v1/d18-2012.

Sayali Kulkarni, Sheide Chammas, Wan Zhu, Fei Sha, and Eugene Ie. Aquamuse: Automatically generating datasets for query-based multi-document summarization. CoRR, abs/2010.12694, 2020. URL https://arxiv.org/abs/2010.12694.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. Data augmentation using pre-trained transformer models. CoRR, abs/2003.02245, 2020. URL https://arxiv.org/abs/2003.02245.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. Summac: Re-visiting nli-based models for inconsistency detection in summarization. Trans. Assoc. Comput. Linguistics, 10:163–177, 2022. doi: 10.1162/TACL_A_00453. URL https://doi.org/10.1162/tacl_a_00453.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning. CoRR, abs/1910.09700, 2019. URL http://arxiv.org/abs/1910.09700.

André Lage-Freitas, Héctor Allende-Cid, Orivaldo Santana, and Lívia de Oliveira-Lage. Predicting brazilian court decisions. PeerJ Comput. Sci., 8:e904, 2022. doi: 10.7717/PEERJ-CS.904. URL https://doi.org/10.7717/peerj-cs.904.

Tuan Manh Lai, Giuseppe Castellucci, Saar Kuzi, Heng Ji, and Oleg Rokhlenko. External knowledge acquisition for end-to-end document-oriented dialog systems. In Andreas Vlachos and Isabelle Augenstein, editors, Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023, pages 3615–3629. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EACL-MAIN.264. URL https://doi.org/10.18653/v1/2023.eacl-main.264.

Yash Kumar Lal, Reetu Singh, Harsh Trivedi, Qingqing Cao, Aruna Balasubramanian, and Niranjan Balasubramanian. Irene-viz: Visualizing energy consumption of transformer models. In Heike Adel and Shuming Shi, editors, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2021, Online and Punta Cana, Dominican Republic, 7-11 November, 2021, pages 251–258. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EMNLP-DEMO.29. URL https://doi.org/10.18653/v1/2021.emnlp-demo.29.

Rémi Leblond, Jean-Baptiste Alayrac, Laurent Sifre, Miruna Pislar, Jean-Baptiste Lespiau, Ioannis Antonoglou, Karen Simonyan, and Oriol Vinyals. Machine translation decoding beyond beam search. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, pages 8410–8434. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EMNLP-MAIN.662. URL https://doi.org/10.18653/v1/2021.emnlp-main.662.

Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In David A. Forsyth, Joseph L. Mundy, Vito Di Gesù, and Roberto Cipolla, editors, Shape, Contour and Grouping in Computer Vision, volume 1681 of Lecture Notes in Computer Science, page 319. Springer, 1999. doi: 10.1007/3-540-46805-6_19. URL https://doi.org/10.1007/3-540-46805-6_19.

Kenton Lee, Kelvin Guu, Luheng He, Tim Dozat, and Hyung Won Chung. Neural data augmentation via example extrapolation. CoRR, abs/2102.01335, 2021a. URL https://arxiv.org/abs/2102.01335.

Minwoo Lee, Seungpil Won, Juae Kim, Hwanhee Lee, Cheon-Eum Park, and Kyomin Jung. Crossaug: A contrastive data augmentation method for debiasing fact verification models. In Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong, editors, CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021, pages 3181–3185. ACM, 2021b. doi: 10.1145/3459637.3482078. URL https://doi.org/10.1145/3459637.3482078.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pages 7871–7880. Association for

Computational Linguistics, 2020a. doi: 10.18653/V1/2020.ACL-MAIN.703. URL https://doi.org/10.18653/v1/2020.acl-main.703.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020b. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Sasko, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander M. Rush, and Thomas Wolf. Datasets: A community library for natural language processing. In Heike Adel and Shuming Shi, editors, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2021, Online and Punta Cana, Dominican Republic, 7-11 November, 2021, pages 175–184. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EMNLP-DEMO.21. URL https://doi.org/10.18653/v1/2021.emnlp-demo.21.

Chenliang Li, Weiran Xu, Si Li, and Sheng Gao. Guiding generation for abstractive text summarization based on key information guide network. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers), pages 55–60. Association for Computational Linguistics, 2018. doi: 10.18653/V1/N18-2009. URL https://doi.org/10.18653/v1/n18-2009.

Da Li, Xinbo Chen, Michela Becchi, and Ziliang Zong. Evaluating the energy efficiency of deep convolutional neural networks on cpus and gpus. In Zhipeng Cai, Rafal A. Angryk, Wen-Zhan Song, Yingshu Li, Xiaojun Cao, Anu G. Bourgeois, Guangchun Luo, Liang Cheng, and Bhaskar Krishnamachari, editors, 2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom), BDCloud-SocialCom-SustainCom 2016, Atlanta, GA, USA, October 8-10, 2016, pages 477–484. IEEE Computer Society, 2016. doi: 10.1109/BDCLOUD-SOCIALCOM-SUSTAINCOM.2016.76. URL https://doi.org/10.1109/BDCloud-SocialCom-SustainCom.2016.76.

Daniel Li, Thomas Chen, Albert Tung, and Lydia B. Chilton. Hierarchical summarization for longform spoken dialog. In Jeffrey Nichols, Ranjitha Kumar, and Michael Nebeling, editors, UIST '21: The 34th Annual ACM Symposium on User Interface Software and Technology, Virtual Event, USA, October 10-14, 2021, pages 582–597. ACM, 2021a. doi: 10.1145/3472749.3474771. URL https://doi.org/10.1145/3472749.3474771.

Haoran Li, Arash Einolghozati, Srinivasan Iyer, Bhargavi Paranjape, Yashar Mehdad, Sonal Gupta, and Marjan Ghazvininejad. EASE: extractive-abstractive summarization with explanations. CoRR, abs/2105.06982, 2021b. URL https://arxiv.org/abs/2105.06982.

Miao Li, Jianzhong Qi, and Jey Han Lau. Compressed heterogeneous graph for abstractive multi-document summarization. In Brian Williams, Yiling Chen, and Jennifer Neville, editors, Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023, pages 13085–13093. AAAI Press, 2023. doi: 10.1609/AAAI.V37I11.26537. URL https://doi.org/10.1609/aaai.v37i11.26537.

Wei Li, Xinyan Xiao, Jiachen Liu, Hua Wu, Haifeng Wang, and Junping Du. Leveraging graph to improve abstractive multi-document summarization. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pages 6232–6243. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.ACL-MAIN.555. URL https://doi.org/10.18653/v1/2020.acl-main.555.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, pages 4582–4597. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.ACL-LONG.353. URL https://doi.org/10.18653/v1/2021.acl-long.353.

Paul Pu Liang, Chiyu Wu, Louis-Philippe Morency, and Ruslan Salakhutdinov. Towards understanding and mitigating social biases in language models. In Marina Meila and Tong Zhang, editors, Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event, volume 139 of Proceedings of Machine Learning Research, pages 6565–6576. PMLR, 2021. URL http://proceedings.mlr.press/v139/liang21a.html.

Kexin Liao, Logan Lebanoff, and Fei Liu. Abstract meaning representation for multi-document summarization. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018, pages 1178–1190. Association for Computational Linguistics, 2018. URL https://aclanthology.org/C18-1101/.

Daniele Licari and Giovanni Comandé. ITALIAN-LEGAL-BERT: A pre-trained transformer language model for italian law. In Danai Symeonidou, Ran Yu, Davide Ceolin, María Poveda-Villalón, Davide Audrito, Luigi Di Caro, Francesca Grasso, Roberto Nai, Emilio Sulis, Fajar J. Ekaputra, Oliver Kutz, and Nicolas Troquard, editors, Companion Proceedings of the 23rd International Conference on Knowledge Engineering and Knowledge Management, Bozen-Bolzano, Italy, September 26-29, 2022, volume 3256 of CEUR Workshop Proceedings. CEUR-WS.org, 2022. URL https://ceur-ws.org/Vol-3256/km4law3.pdf.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In Text Summarization Branches Out, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013.

Jimmy Lin, Rodrigo Frassetto Nogueira, and Andrew Yates. Pretrained Transformers for Text Ranking: BERT and Beyond. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2021. ISBN 978-3-031-01053-8. doi: 10.2200/S01123ED1V01Y202108HLT053. URL https://doi.org/10.2200/S01123ED1V01Y202108HLT053.

Michael Lissner and Brian W. Carver. Courtlistener.com: A platform for researching and staying abreast of the latest law. http://www.courtlistener.com, 2010.

Jiacheng Liu, Wenya Wang, Dianzhuo Wang, Noah A. Smith, Yejin Choi, and Hannaneh Hajishirzi. Vera: A general-purpose plausibility estimation model for commonsense statements. CoRR, abs/2305.03695, 2023. doi: 10.48550/ARXIV.2305.03695. URL https://doi.org/10.48550/arXiv.2305.03695.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018. URL https://openreview.net/forum?id=Hyg0vbWC-.

Shuaiqi Liu, Jiannong Cao, Ruosong Yang, and Zhiyuan Wen. Generating a structured summary of numerous academic papers: Dataset and method. In Luc De Raedt, editor, Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022, pages 4259–4265. ijcai.org, 2022a. doi: 10.24963/IJCAI.2022/591. URL https://doi.org/10.24963/ijcai.2022/591.

Xiaochen Liu, Yang Gao, Yu Bai, Jiawei Li, Yinan Hu, Heyan Huang, and Boxing Chen. PSP: pre-trained soft prompts for few-shot abstractive summarization. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022, pages 6355–6368. International Committee on Computational Linguistics, 2022b. URL https://aclanthology.org/2022.coling-1.553.

Yang Liu. Fine-tune BERT for extractive summarization. CoRR, abs/1903.10318, 2019. URL http://arxiv.org/abs/1903.10318.

Yang Liu and Mirella Lapata. Hierarchical transformers for multi-document summarization. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pages 5070–5081. Association for Computational Linguistics, 2019a. doi: 10.18653/V1/P19-1500. URL https://doi.org/10.18653/v1/p19-1500.

Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pages 3728–3738. Association for Computational Linguistics, 2019b. doi: 10.18653/V1/D19-1387. URL https://doi.org/10.18653/v1/D19-1387.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. CoRR, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. Trans. Assoc. Comput. Linguistics, 8:726–742, 2020. doi: 10.1162/TACL_A_00343. URL https://doi.org/10.1162/tacl_a_00343.

Yixin Liu, Ansong Ni, Linyong Nan, Budhaditya Deb, Chenguang Zhu, Ahmed Hassan Awadallah, and Dragomir Radev. Leveraging locality in abstractive text summarization. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, pages 6081–6093. Association for Computational Linguistics, 2022c. doi: 10.18653/V1/2022.EMNLP-MAIN.408. URL https://doi.org/10.18653/v1/2022.emnlp-main.408.

Zhengyuan Liu and Nancy Chen. Exploiting discourse-level segmentation for extractive summarization. In Lu Wang, Jackie Chi Kit Cheung, Giuseppe Carenini, and Fei Liu, editors, Proceedings of the 2nd Workshop on New Frontiers in Summarization, pages 116–121, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5415. URL https://aclanthology.org/D19-5415.

Stefano Lodi, Gianluca Moro, and Claudio Sartori. Distributed data clustering in multi-dimensional peer-to-peer networks. In Heng Tao Shen and Athman Bouguettaya, editors, Database Technologies 2010,

Twenty-First Australasian Database Conference (ADC 2010), Brisbane, Australia, 18-22 January, 2010, Proceedings, volume 104 of CRPIT, pages 171–178. Australian Computer Society, 2010. URL http://portal.acm.org/citation.cfm?id=1862264&CFID=17470975&CFTOKEN=71845406.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Kadan Lottick, Silvia Susai, Sorelle A. Friedler, and Jonathan P. Wilson. Energy usage reports: Environmental awareness as part of algorithmic accountability. CoRR, abs/1911.08354, 2019. URL http://arxiv.org/abs/1911.08354.

Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. Interpretable long-form legal question answering with retrieval-augmented large language models. CoRR, abs/2309.17050, 2023. doi: 10.48550/ARXIV.2309.17050. URL https://doi.org/10.48550/arXiv.2309.17050.

Jordan J Louviere and George G Woodworth. Best-worst scaling: A model for the largest difference judgments. Technical report, Working paper, 1991.

Jordan J Louviere, Terry N Flynn, and Anthony Alfred John Marley. Best-worst scaling: Theory, methods and applications. Cambridge University Press, 2015.

Bruce T Lowerre. The harpy speech recognition system. Carnegie Mellon University, 1976.

Michal Lukasik, Boris Dadachev, Kishore Papineni, and Gonçalo Simões. Text segmentation by cross segment attention. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pages 4707–4716. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.EMNLP-MAIN.380. URL https://doi.org/10.18653/v1/2020.emnlp-main.380.

Congbo Ma, Wei Emma Zhang, Mingyu Guo, Hu Wang, and Quan Z. Sheng. Multi-document summarization via deep learning techniques: A survey. ACM Comput. Surv., 55(5):102:1–102:37, 2023. doi: 10.1145/3529754. URL https://doi.org/10.1145/3529754.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA, pages 142–150. The Association for Computer Linguistics, 2011. URL https://aclanthology.org/P11-1015/.

Ahmed Magooda and Diane J. Litman. Abstractive summarization for low resource data using domain transfer and data synthesis. In Roman Barták and Eric Bell, editors, Proceedings of the Thirty-Third International Florida Artificial Intelligence Research Society Conference, Originally to be held in North Miami Beach, Florida, USA, May 17-20, 2020, pages 240–245. AAAI Press, 2020. URL https://aaai.org/ocs/index.php/FLAIRS/FLAIRS20/paper/view/18441.

Debanjan Mahata, Navneet Agarwal, Dibya Gautam, Amardeep Kumar, Swapnil Parekh, Yaman Kumar Singla, Anish Acharya, and Rajiv Ratn Shah. LDKP - A dataset for identifying keyphrases from long scientific documents. In Wei Liu and Linsey Pang, editors, Proceedings of the Workshop on Deep Learning for Search and Recommendation (DL4SR 2022) co-located with the 31st ACM International Conference on Information and Knowledge Management (CIKM 2022), Atlanta, Georgia, USA, October 17-21, 2022, volume 3317 of CEUR Workshop Proceedings. CEUR-WS.org, 2022. URL https://ceur-ws.org/Vol-3317/Paper9.pdf.

Chanakya Malireddy, Tirth Maniar, and Manish Shrivastava. SCAR: sentence compression using autoencoders for reconstruction. In Shruti Rijhwani, Jiangming Liu, Yizhong Wang, and Rotem Dror, editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, ACL 2020, Online, July 5-10, 2020, pages 88–94. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.ACL-SRW.13. URL https://doi.org/10.18653/v1/2020.acl-srw.13.

Potsawee Manakul and Mark J. F. Gales. Long-span summarization via local attention and content selection. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, pages 6026–6041. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.ACL-LONG.470. URL https://doi.org/10.18653/v1/2021.acl-long.470.

Supun S. Manathunga and Y. A. Illangasekara. Retrieval augmented generation and representative vector summarization for large unstructured textual data in medical education. CoRR, abs/2308.00479, 2023. doi: 10.48550/ARXIV.2308.00479. URL https://doi.org/10.48550/arXiv.2308.00479.

Arpan Mandal, Kripabandhu Ghosh, Arindam Pal, and Saptarshi Ghosh. Automatic catchphrase identification from legal court case documents. In Ee-Peng Lim, Marianne Winslett, Mark Sanderson, Ada Wai-Chee Fu, Jimeng Sun, J. Shane Culpepper, Eric Lo, Joyce C. Ho, Debora Donato, Rakesh Agrawal, Yu Zheng, Carlos Castillo, Aixin Sun, Vincent S. Tseng, and Chenliang Li, editors, Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017, pages 2187–2190. ACM, 2017. doi: 10.1145/3132847.3133102. URL https://doi.org/10.1145/3132847.3133102.

Ziming Mao, Chen Henry Wu, Ansong Ni, Yusen Zhang, Rui Zhang, Tao Yu, Budhaditya Deb, Chenguang Zhu, Ahmed Hassan Awadallah, and Dragomir R. Radev. DYLE: dynamic latent extraction for abstractive long-input summarization. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 1687–1698. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.ACL-LONG.118. URL https://doi.org/10.18653/v1/2022.acl-long.118.

André F. T. Martins, António Farinhas, Marcos V. Treviso, Vlad Niculae, Pedro M. Q. Aguiar, and Mário A. T. Figueiredo. Sparse and continuous attention mechanisms. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/f0b76267fbe12b936bd65e203dc675c1-Abstract.html.

Pedro Henrique Martins, Zita Marinho, and André F. T. Martins. ∞-former: Infinite memory transformer. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 5468–5485. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.ACL-LONG.375. URL https://doi.org/10.18653/v1/2022.acl-long.375.

Ayushi Mathur and M Suchithra. Application of abstractive summarization in multiple choice question generation. In 2022 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), pages 409–413, 2022. doi: 10.1109/CISES54857.2022.9844396.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December

4-9, 2017, Long Beach, CA, USA, pages 6294–6305, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/20c86a628232a67e7bd46f76fba7ce12-Abstract.html.

Clara Meister, Ryan Cotterell, and Tim Vieira. Best-first beam search. Trans. Assoc. Comput. Linguistics, 8:795–809, 2020. doi: 10.1162/TACL_A_00346. URL https://doi.org/10.1162/tacl_a_00346.

Clara Meister, Gian Wiher, and Ryan Cotterell. On decoding strategies for neural text generators. Trans. Assoc. Comput. Linguistics, 10:997–1012, 2022. URL https://transacl.org/ojs/index.php/tacl/article/view/3807.

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. Locally typical sampling. Trans. Assoc. Comput. Linguistics, 11:102–121, 2023. URL https://transacl.org/ojs/index.php/tacl/article/view/3993.

Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. Weakly-supervised hierarchical text classification. In The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, pages 6826–6833. AAAI Press, 2019. doi: 10.1609/AAAI.V33I01.33016826. URL https://doi.org/10.1609/aaai.v33i01.33016826.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. Generating training data with language models: Towards zero-shot language understanding. In NeurIPS, 2022a. URL http://papers.nips.cc/paper_files/paper/2022/hash/0346c148ba1c21c6b4780a961ea141dc-Abstract-Conference.html.

Zaiqiao Meng, Fangyu Liu, Ehsan Shareghi, Yixuan Su, Charlotte Collins, and Nigel Collier. Rewire-then-probe: A contrastive recipe for probing biomedical knowledge of pre-trained language models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 4798–4810. Association for Computational Linguistics, 2022b. doi: 10.18653/V1/2022.ACL-LONG.329. URL https://doi.org/10.18653/v1/2022.acl-long.329.

Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain, pages 404–411. ACL, 2004. URL https://www.aclweb.org/anthology/W04-3252/.

George A. Miller. Wordnet: A lexical database for english. Commun. ACM, 38(11):39–41, 1995. doi: 10.1145/219717.219748. URL https://doi.org/10.1145/219717.219748.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. Recent advances in natural language processing via large pre-trained language models: A survey. ACM Comput. Surv., 56(2):30:1–30:40, 2024. doi: 10.1145/3605943. URL https://doi.org/10.1145/3605943.

Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andrés Codas, Clarisse Simões, Sahaj Agrawal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, Hamid Palangi, Guoqing Zheng, Corby Rosset, Hamed Khanpour, and Ahmed Hassan Awadallah. Orca 2: Teaching small language models how to reason. CoRR, abs/2311.11045, 2023. doi: 10.48550/ARXIV.2311.11045. URL https://doi.org/10.48550/arXiv.2311.11045.

Timo Möller, Anthony Reina, Raghavan Jayakumar, and Malte Pietsch. COVID-QA: A question answering dataset for COVID-19. July 2020. URL https://aclanthology.org/2020.nlpcovid19-acl.18.

Gabriele Monti and Gianluca Moro. Multidimensional range query and load balancing in wireless ad hoc and sensor networks. In Klaus Wehrle, Wolfgang Kellerer, Sandeep K. Singhal, and Ralf Steinmetz, editors, Proceedings P2P'08, Eighth International Conference on Peer-to-Peer Computing, 8-11 September 2008, Aachen, Germany, pages 205–214. IEEE Computer Society, 2008. doi: 10.1109/P2P.2008.27. URL https://doi.org/10.1109/P2P.2008.27.

Gianluca Moro and Marco Masseroli. Gene function finding through cross-organism ensemble learning. BioData Min., 14(1):14, 2021. doi: 10.1186/S13040-021-00239-W. URL https://doi.org/10.1186/s13040-021-00239-w.

Gianluca Moro and Gabriele Monti. W-grid: A scalable and efficient self-organizing infrastructure for multi-dimensional data management, querying and routing in wireless data-centric sensor networks. J. Netw. Comput. Appl., 35(4):1218–1234, 2012. doi: 10.1016/J.JNCA.2011.05.002. URL https://doi.org/10.1016/j.jnca.2011.05.002.

Gianluca Moro and Luca Ragazzi. Semantic self-segmentation for abstractive summarization of long documents in low-resource regimes. In Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022, pages 11085–11093. AAAI Press, 2022. doi: 10.1609/AAAI.V36I10.21357. URL https://doi.org/10.1609/aaai.v36i10.21357.

Gianluca Moro and Luca Ragazzi. Align-then-abstract representation learning for low-resource summarization. Neurocomputing, 548:126356, 2023. doi: 10.1016/J.NEUCOM.2023.126356. URL https://doi.org/10.1016/j.neucom.2023.126356.

Gianluca Moro and Stefano Salvatori. Deep vision-language model for efficient multi-modal similarity search in fashion retrieval. In Tomás Skopal, Fabrizio Falchi, Jakub Lokoc, Maria Luisa Sapino, Ilaria Bartolini, and Marco Patella, editors, Similarity Search and Applications - 15th International Conference, SISAP 2022, Bologna, Italy, October 5-7, 2022, Proceedings, volume 13590 of Lecture Notes in Computer Science, pages 40–53. Springer, 2022. doi: 10.1007/978-3-031-17849-8_4. URL https://doi.org/10.1007/978-3-031-17849-8_4.

Gianluca Moro and Lorenzo Valgimigli. Efficient self-supervised metric information retrieval: A bibliography based method applied to COVID literature. Sensors, 21(19):6430, 2021. doi: 10.3390/S21196430. URL https://doi.org/10.3390/s21196430.

Gianluca Moro, Andrea Pagliarani, Roberto Pasolini, and Claudio Sartori. Cross-domain & in-domain sentiment analysis with memory-based deep neural networks. In Ana L. N. Fred and Joaquim Filipe, editors, Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K 2018, Volume 1: KDIR, Seville, Spain, September 18-20, 2018, pages 125–136. SciTePress, 2018. doi: 10.5220/0007239101270138. URL https://doi.org/10.5220/0007239101270138.

Gianluca Moro, Luca Ragazzi, Lorenzo Valgimigli, and Davide Freddi. Discriminative marginalized probabilistic neural method for multi-document summarization of medical literature. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 180–189. Association for Computational Linguistics, 2022a. doi: 10.18653/V1/2022.ACL-LONG.15. URL https://doi.org/10.18653/v1/2022.acl-long.15.

Gianluca Moro, Lorenzo Valgimigli, Alex Rossi, Cristiano Casadei, and Andrea Montefiori. Self-supervised information retrieval trained from self-generated sets of queries and relevant documents. In Tomás Skopal, Fabrizio Falchi, Jakub Lokoc, Maria Luisa Sapino, Ilaria Bartolini, and Marco Patella, editors, Similarity Search and Applications - 15th International Conference,

SISAP 2022, Bologna, Italy, October 5-7, 2022, Proceedings, volume 13590 of Lecture Notes in Computer Science, pages 283–290. Springer, 2022b. doi: 10.1007/978-3-031-17849-8_23. URL https://doi.org/10.1007/978-3-031-17849-8_23.

Gianluca Moro, Nicola Piscaglia, Luca Ragazzi, and Paolo Italiani. Multi-language transfer learning for low-resource legal case summarization. Artificial Intelligence and Law, pages 1–29, 2023a. doi: 10.1007/s10506-023-09373-8. URL https://doi.org/10.1007/s10506-023-09373-8.

Gianluca Moro, Luca Ragazzi, and Lorenzo Valgimigli. Carburacy: Summarization models tuning and comparison in eco-sustainable regimes with a novel carbon-aware accuracy. In Brian Williams, Yiling Chen, and Jennifer Neville, editors, Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023, pages 14417–14425. AAAI Press, 2023b. doi: 10.1609/AAAI.V37I12.26686. URL https://doi.org/10.1609/aaai.v37i12.26686.

Gianluca Moro, Luca Ragazzi, and Lorenzo Valgimigli. Graph-based abstractive summarization of extracted essential knowledge for low-resource scenarios. In Kobi Gal, Ann Nowé, Grzegorz J. Nalepa, Roy Fairstein, and Roxana Radulescu, editors, ECAI 2023 - 26th European Conference on Artificial Intelligence, September 30 - October 4, 2023, Kraków, Poland - Including 12th Conference on Prestigious Applications of Intelligent Systems (PAIS 2023), volume 372 of Frontiers in Artificial Intelligence and Applications, pages 1747–1754. IOS Press, 2023c. doi: 10.3233/FAIA230460. URL https://doi.org/10.3233/FAIA230460.

Gianluca Moro, Luca Ragazzi, Lorenzo Valgimigli, Giacomo Frisoni, Claudio Sartori, and Gustavo Marfia. Efficient memory-enhanced transformer for long-document summarization in low-resource regimes. Sensors, 23(7):3542, 2023d. doi: 10.3390/S23073542. URL https://doi.org/10.3390/s23073542.

Gianluca Moro, Luca Ragazzi, Lorenzo Valgimigli, and Lorenzo Molfetta. Retrieve-and-rank end-to-end summarization of biomedical studies. In Oscar Pedreira and Vladimir Estivill-Castro, editors, Similarity Search and Applications - 16th International Conference, SISAP 2023, A Coruña, Spain, October 9-11, 2023, Proceedings, volume 14289 of Lecture Notes in Computer Science, pages 64–78. Springer, 2023e. doi: 10.1007/978-3-031-46994-7_6. URL https://doi.org/10.1007/978-3-031-46994-7_6.

Gianluca Moro, Stefano Salvatori, and Giacomo Frisoni. Efficient text-image semantic search: A multi-modal vision-language approach for fashion retrieval. Neurocomputing, 538:126196, 2023f. doi: 10.1016/J.NEUCOM.2023.03.057. URL https://doi.org/10.1016/j.neucom.2023.03.057.

Gianluca Moro, Luca Ragazzi, Lorenzo Valgimigli, Fabian Vincenzi, and Davide Freddi. Revelio: Interpretable long-form question answering. In The Second Tiny Papers Track at ICLR 2024, Tiny Papers @ ICLR 2024, Vienna, Austria, May 7-11, 2024. OpenReview.net, 2024.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M. Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. Crosslingual generalization through multitask finetuning. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023, pages 15991–16111. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.891. URL https://doi.org/10.18653/v1/2023.acl-long.891.

Rajdeep Mukherjee, Abhinav Bohra, Akash Banerjee, Soumya Sharma, Manjunath Hegde, Afreen Shaikh, Shivani Shrivastava, Koustuv Dasgupta, Niloy Ganguly, Saptarshi Ghosh, and Pawan Goyal.

Ectsum: A new benchmark dataset for bullet point summarization of long earnings call transcripts. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, pages 10893–10906. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.748. URL https://doi.org/10.18653/v1/2022.emnlp-main.748.

Donatella Muratore, Markus Hagenbuchner, Franco Scarselli, and Ah Chung Tsoi. Sentence extraction by graph neural networks. In Konstantinos I. Diamantaras, Wlodek Duch, and Lazaros S. Iliadis, editors, Artificial Neural Networks - ICANN 2010 - 20th International Conference, Thessaloniki, Greece, September 15-18, 2010, Proceedings, Part III, volume 6354 of Lecture Notes in Computer Science, pages 237–246. Springer, 2010. doi: 10.1007/978-3-642-15825-4_29. URL https://doi.org/10.1007/978-3-642-15825-4_29.

Rakshit Naidu, Harshita Diddee, Ajinkya Mulay, Aleti Vardhan, Krithika Ramesh, and Ahmed S. Zamzam. Towards quantifying the carbon emissions of differentially private machine learning. CoRR, abs/2107.06946, 2021. URL https://arxiv.org/abs/2107.06946.

Preslav Nakov, David P. A. Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, Alberto Barrón-Cedeño, Paolo Papotti, Shaden Shaar, and Giovanni Da San Martino. Automated fact-checking for assisting human fact-checkers. In Zhi-Hua Zhou, editor, Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021, pages 4551–4558. ijcai.org, 2021. doi: 10.24963/IJCAI.2021/619. URL https://doi.org/10.24963/ijcai.2021/619.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence rnns and beyond. In Yoav Goldberg and Stefan Riezler, editors, Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016, pages 280–290. ACL, 2016. doi: 10.18653/V1/K16-1028. URL https://doi.org/10.18653/v1/k16-1028.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, pages 1797–1807. Association for Computational Linguistics, 2018. doi: 10.18653/V1/D18-1206. URL https://doi.org/10.18653/v1/d18-1206.

Mir Tafseer Nayeem, Tanvir Ahmed Fuad, and Yllias Chali. Abstractive unsupervised multi-document summarization using paraphrastic sentence fusion. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018, pages 1191–1204. Association for Computational Linguistics, 2018. URL https://aclanthology.org/C18-1102/.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human generated machine reading comprehension dataset. In Tarek Richard Besold, Antoine Bordes, Artur S. d'Avila Garcez, and Greg Wayne, editors, Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016, volume 1773 of CEUR Workshop Proceedings. CEUR-WS.org, 2016. URL https://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf.

Joel Niklaus, Veton Matoshi, Pooja Rani, Andrea Galassi, Matthias Stürmer, and Ilias Chalkidis. LEXTREME: A multi-lingual and multi-task benchmark for the legal domain. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, Findings of the Association for Computational Linguistics: EMNLP

2023, Singapore, December 6-10, 2023, pages 3016–3054. Association for Computational Linguistics, 2023a. URL https://aclanthology.org/2023.findings-emnlp.200.

Joel Niklaus, Veton Matoshi, Matthias Stürmer, Ilias Chalkidis, and Daniel E. Ho. Multilegalpile: A 689gb multilingual legal corpus. CoRR, abs/2306.02069, 2023b. doi: 10.48550/ARXIV.2306.02069. URL https://doi.org/10.48550/arXiv.2306.02069.

Narjes Nikzad-Khasmakhi, Mohammad-Reza Feizi-Derakhshi, Meysam Asgari-Chenaghlu, Mohammad Ali Balafar, Ali-Reza Feizi-Derakhshi, Taymaz Rahkar-Farshi, Majid Ramezani, Zoleikha Jahanbakhsh-Nagadeh, Elnaz Zafarani-Moattar, and Mehrdad Ranjbar-Khadivi. Phraseformer: Multimodal key-phrase extraction using transformer and graph embedding. CoRR, abs/2106.04939, 2021. URL https://arxiv.org/abs/2106.04939.

OpenAI. GPT-4 technical report. CoRR, abs/2303.08774, 2023. doi: 10.48550/ARXIV.2303.08774. URL https://doi.org/10.48550/arXiv.2303.08774.

Feifei Pan, Mustafa Canim, Michael R. Glass, Alfio Gliozzo, and James A. Hendler. End-to-end table question answering via retrieval-augmented generation. CoRR, abs/2203.16714, 2022. doi: 10.48550/ARXIV.2203.16714. URL https://doi.org/10.48550/arXiv.2203.16714.

Liangming Pan, Wenhu Chen, Wenhan Xiong, Min-Yen Kan, and William Yang Wang. Zero-shot fact verification by claim generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, August 1-6, 2021, pages 476–483. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-short.61. URL https://doi.org/10.18653/v1/2021.acl-short.61.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. CoRR, abs/2306.08302, 2023. doi: 10.48550/ARXIV.2306.08302. URL https://doi.org/10.48550/arXiv.2306.08302.

Yannis Papanikolaou, Marlene Staib, Justin Joshua Grace, and Francine Bennett. Slot filling for biomedical information extraction. In Dina Demner-Fushman, Kevin Bretonnel Cohen, Sophia Ananiadou, and Junichi Tsujii, editors, Proceedings of the 21st Workshop on Biomedical Language Processing, BioNLP@ACL 2022, Dublin, Ireland, May 26, 2022, pages 82–90. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.BIONLP-1.7. URL https://doi.org/10.18653/v1/2022.bionlp-1.7.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA, pages 311–318. ACL, 2002. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

Shantipriya Parida and Petr Motlícek. Abstract text summarization: A low resource challenge. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pages 5993–5997. Association for Computational Linguistics, 2019. doi: 10.18653/V1/D19-1616. URL https://doi.org/10.18653/v1/D19-1616.

Ankur P. Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. Totto: A controlled table-to-text generation dataset. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pages 1173–1186. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.EMNLP-MAIN.89. URL https://doi.org/10.18653/v1/2020.emnlp-main.89.

Ramakanth Pasunuru and Mohit Bansal. Multi-reward reinforced summarization with saliency and entailment. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers), pages 646–653. Association for Computational Linguistics, 2018. doi: 10.18653/v1/n18-2102. URL https://doi.org/10.18653/v1/n18-2102.

Ramakanth Pasunuru, Mengwen Liu, Mohit Bansal, Sujith Ravi, and Markus Dreyer. Efficiently summarizing text and graph encodings of multi-document clusters. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021, pages 4768–4779. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.NAACL-MAIN.380. URL https://doi.org/10.18653/v1/2021.naacl-main.380.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pages 8024–8035, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

David A. Patterson, Joseph Gonzalez, Urs Hölzle, Quoc V. Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David R. So, Maud Texier, and Jeff Dean. The carbon footprint of machine learning training will plateau, then shrink. Computer, 55(7):18–28, 2022. doi: 10.1109/MC.2022.3148714. URL https://doi.org/10.1109/MC.2022.3148714.

Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018. URL https://openreview.net/forum?id=HkAClQgA-.

Keqin Peng, Liang Ding, Qihuang Zhong, Li Shen, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. Towards making the most of chatgpt for machine translation. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023, pages 5622–5633. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.findings-emnlp.373.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers), pages 2227–2237. Association for Computational Linguistics, 2018. doi: 10.18653/v1/n18-1202. URL https://doi.org/10.18653/v1/n18-1202.

Jason Phang, Yao Zhao, and Peter J. Liu. Investigating efficiently extending transformers for long input summarization. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023, pages 3946–3961. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.emnlp-main.240.

Jonathan Pilault, Raymond Li, Sandeep Subramanian, and Chris Pal. On extractive and abstractive neural document summarization with transformer language models. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pages 9308–9319. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.748. URL https://doi.org/10.18653/v1/2020.emnlp-main.748.

Barbara Plank. What to do about non-standard (or non-canonical) language in NLP. In Stefanie Dipper, Friedrich Neubarth, and Heike Zinsmeister, editors, Proceedings of the 13th Conference on Natural Language Processing, KONVENS 2016, Bochum, Germany, September 19-21, 2016, volume 16 of Bochumer Linguistische Arbeitsberichte, 2016. URL https://www.linguistics.rub.de/konvens16/pub/2_konvensproc.pdf.

Adam Poliak, Max Fleming, Cash Costello, Kenton W. Murray, Mahsa Yarmohammadi, Shivani Pandya, Darius Irani, Milind Agarwal, Udit Sharma, Shuo Sun, Nicola Ivanov, Lingxi Shang, Kaushik Srinivasan, Seolhwa Lee, Xu Han, Smisha Agarwal, and João Sedoc. Collecting verified COVID-19 question answer pairs. In Karin Verspoor, Kevin Bretonnel Cohen, Michael Conway, Berry de Bruijn, Mark Dredze, Rada Mihalcea, and Byron C. Wallace, editors, Proceedings of the 1st Workshop on NLP for COVID-19@ EMNLP 2020, Online, December 2020. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.NLPCOVID19-2.31. URL https://doi.org/10.18653/v1/2020.nlpcovid19-2.31.

Raul Puri, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. Training question answering models from synthetic data. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pages 5811–5826. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.EMNLP-MAIN.468. URL https://doi.org/10.18653/v1/2020.emnlp-main.468.

Hans O Pörtner, Debra C Roberts, Helen Adams, Carolina Adler, et al. Climate change 2022: impacts, adaptation and vulnerability. IPCC, 2022.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. In Trevor Cohn, Yulan He, and Yang Liu, editors, Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020, volume EMNLP 2020 of Findings of ACL, pages 2401–2410. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.FINDINGS-EMNLP.217. URL https://doi.org/10.18653/v1/2020.findings-emnlp.217.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. Pre-trained models for natural language processing: A survey. CoRR, abs/2003.08271, 2020. URL https://arxiv.org/abs/2003.08271.

Yifu Qiu and Shay B. Cohen. Abstractive summarization guided by latent hierarchical document structure. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, pages 5303–5317. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.355. URL https://doi.org/10.18653/v1/2022.emnlp-main.355.

Dragomir R. Radev. A common theory of information fusion from multiple text sources step one: Cross-document structure. In Laila Dybkjær, Kôiti Hasida, and David R. Traum, editors, Proceedings of the SIGDIAL 2000 Workshop, The 1st Annual Meeting of the Special Interest Group on Discourse and Dialogue, 7-8 October 2000, Hong Kong, pages 74–83. The Association for Computer Linguistics, 2000. doi: 10.3115/1117736.1117745. URL https://aclanthology.org/W00-1009/.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. OpenAI Blog, 1(8):9, 2019a.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9, 2019b.

Jack W. Rae and Ali Razavi. Do transformers need deep long-range memory? In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pages 7524–7529. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.ACL-MAIN.672. URL https://doi.org/10.18653/v1/2020.acl-main.672.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020. URL https://openreview.net/forum?id=SylKikSYDH.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res., 21:140:1–140:67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Luca Ragazzi, Giacomo Frisoni, Gianluca Moro, Paolo Italiani, Lorenzo Molfetta, and Veronika Folin. COMMA: A multi-task and multi-lingual dataset of constitutional verdicts, 2024. Submitted to Data-centric Machine Learning Research.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. In Jian Su, Xavier Carreras, and Kevin Duh, editors, Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016, pages 2383–2392. The Association for Computational Linguistics, 2016. doi: 10.18653/V1/D16-1264. URL https://doi.org/10.18653/v1/d16-1264.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. In Iryna Gurevych and Yusuke Miyao, editors, Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers, pages 784–789. Association for Computational Linguistics, 2018. doi: 10.18653/V1/P18-2124. URL https://aclanthology.org/P18-2124/.

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization. In The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net, 2023. URL https://openreview.net/pdf?id=8aHzds2uUyB.

Abhilasha Ravichander, Alan W. Black, Shomir Wilson, Thomas B. Norton, and Norman M. Sadeh. Question answering for privacy policies: Combining computational and legal perspectives. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pages 4946–4957. Association for Computational Linguistics, 2019. doi: 10.18653/V1/D19-1500. URL https://doi.org/10.18653/v1/D19-1500.

Tohida Rehman, Ronit Mandal, Abhishek Agarwal, and Debarshi Kumar Sanyal. Hallucination reduction in long input text summarization. CoRR, abs/2309.16781, 2023. doi: 10.48550/ARXIV.2309.16781. URL https://doi.org/10.48550/arXiv.2309.16781.

Radim Rehurek and Petr Sojka. Gensim–python framework for vector space modelling. NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic, 3(2), 2011.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pages 3980–3990. Association for Computational Linguistics, 2019. doi: 10.18653/V1/D19-1410. URL https://doi.org/10.18653/v1/D19-1410.

Ruiyang Ren, Shangwen Lv, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. PAIR: leveraging passage-centric similarity relation for improving dense passage retrieval. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021, volume ACL/IJCNLP 2021 of Findings of ACL, pages 2173–2183. Association for Computational Linguistics, 2021a. doi: 10.18653/V1/2021.FINDINGS-ACL.191. URL https://doi.org/10.18653/v1/2021.findings-acl.191.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. Rocketqav2: A joint training method for dense passage retrieval and passage re-ranking. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, pages 2825–2835. Association for Computational Linguistics, 2021b. doi: 10.18653/V1/2021.EMNLP-MAIN.224. URL https://doi.org/10.18653/v1/2021.emnlp-main.224.

Stephen E. Robertson and Steve Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In W. Bruce Croft and C. J. van Rijsbergen, editors, Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland, 3-6 July 1994 (Special Issue of the SIGIR Forum), pages 232–241. ACM/Springer, 1994. doi: 10.1007/978-1-4471-2099-5_24. URL https://doi.org/10.1007/978-1-4471-2099-5_24.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. In Donna K. Harman, editor, Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994, volume 500-225 of NIST Special Publication, pages 109–126. National Institute of Standards and Technology (NIST), 1994. URL http://trec.nist.gov/pubs/trec3/papers/city.ps.gz.

Tobias Rohde, Xiaoxia Wu, and Yinhan Liu. Hierarchical learning for generation with long source sequences. CoRR, abs/2104.07545, 2021. URL https://arxiv.org/abs/2104.07545.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. Automatic keyword extraction from individual documents. Text mining: applications and theory, pages 1–20, 2010.

Andrés Rosso-Mateus, Fabio A. González, and Manuel Montes-y-Gómez. A deep metric learning method for biomedical passage retrieval. In Donia Scott, Núria Bel, and Chengqing Zong, editors, Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020, pages 6229–6239. International Committee on Computational Linguistics, 2020. doi: 10.18653/V1/2020.COLING-MAIN.548. URL https://doi.org/10.18653/v1/2020.coling-main.548.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. Leveraging pre-trained checkpoints for sequence generation tasks. Trans. Assoc. Comput. Linguistics, 8:264–280, 2020. doi: 10.1162/TACL_A_00313. URL https://doi.org/10.1162/tacl_a_00313.

Prasenjeet Roy and Suman Kundu. Review on query-focused multi-document summarization (QMDS) with comparative analysis. ACM Comput. Surv., 56(1):5:1–5:38, 2024. doi: 10.1145/3597299. URL https://doi.org/10.1145/3597299.

Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. Covid-fact: Fact extraction and verification of real-world claims on COVID-19 pandemic. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, pages 2116–2129. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.165. URL https://doi.org/10.18653/v1/2021.acl-long.165.

Nipun Sadvilkar and Mark Neumann. Pysbd: Pragmatic sentence boundary disambiguation. CoRR, abs/2010.09657, 2020. URL https://arxiv.org/abs/2010.09657.

Mohammed Saeed, Nicolas Traub, Maelle Nicolas, Gianluca Demartini, and Paolo Papotti. Crowdsourced fact-checking at twitter: How does the crowd compare with experts? In Mohammad Al Hasan and Li Xiong, editors, Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022, pages 1736–1746. ACM, 2022. doi: 10.1145/3511808.3557279. URL https://doi.org/10.1145/3511808.3557279.

Marco Ferdinand Salchner and Adam Jatowt. A survey of automatic text summarization using graph neural networks. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022, pages 6139–6150. International Committee on Computational Linguistics, 2022. URL https://aclanthology.org/2022.coling-1.536.

Evan Sandhaus. The new york times annotated corpus. Linguistic Data Consortium, Philadelphia, 6 (12):e26752, 2008.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. CoRR, abs/1910.01108, 2019. URL http://arxiv.org/abs/1910.01108.

Carlo Sansone and Giancarlo Sperlí. Legal information retrieval systems: State-of-the-art and open issues. Inf. Syst., 106:101967, 2022. doi: 10.1016/J.IS.2021.101967. URL https://doi.org/10.1016/j.is.2021.101967.

Aalok Sathe, Salar Ather, Tuan Manh Le, Nathan Perry, and Joonsuk Park. Automated fact-checking of claims from wikipedia. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, and Stelios Piperidis, editors, Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020, pages 6874–6882. European Language Resources Association, 2020. URL https://aclanthology.org/2020.lrec-1.849/.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel,

Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. BLOOM: A 176b-parameter open-access multilingual language model. CoRR, abs/2211.05100, 2022. doi: 10.48550/ARXIV.2211.05100. URL https://doi.org/10.48550/arXiv.2211.05100.

Timo Schick and Hinrich Schütze. Generating datasets with pretrained language models. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, pages 6943–6951. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EMNLP-MAIN.555. URL https://doi.org/10.18653/v1/2021.emnlp-main.555.

Stephanie Schoch, Diyi Yang, and Yangfeng Ji. "this is a problem, don't you agree?" framing and bias in human evaluation for natural language generation. In Proceedings of the 1st Workshop on Evaluating NLG Evaluation, pages 10–16, Online (Dublin, Ireland), December 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.evalnlgeval-1.2.

John Schulman, Barret Zoph, Christina Kim, Jacob Hilton, Jacob Menick, Jiayi Weng, Juan Felipe Ceron Uribe, Liam Fedus, Luke Metz, Michael Pokorny, Rapha Gontijo Lopes, Shengjia Zhao, Arun Vijayvergiya, Eric Sigler, Adam Perelman, Chelsea Voss, Mike Heaton, Joel Parish, Dave Cummings, Rajeev Nayak, Valerie Balcom, David Schnurr, Tomer Kaftan, Chris Hallacy, Nicholas Turley, Noah Deutsch, Vik Goel, Jonathan Ward, Aris Konstantinidis, Wojciech Zaremba, Long Ouyang, Leonard Bogdonoff, Joshua Gross, David Medina, Sarah Yoo, Teddy Lee, Ryan Lowe, Dan Mossing, Joost Huizinga, Roger Jiang, Carroll Wainwright, Diogo Almeida, Steph Lin, Marvin Zhang, Kai Xiao, Katarina Slama, Steven Bills, Alex Gray, Jan Leike, Jakub Pachocki, Phil Tillet, Shantanu Jain, Greg Brockman, Nick Ryder, Alex Paino, Qiming Yuan, Clemens Winter, Ben Wang, Mo Bavarian, Igor Babuschkin, Szymon Sidor, Ingmar Kanitscheider, Mikhail Pavlov, Matthias Plappert, Nik Tezak, Heewoo Jun, William Zhuk, Vitchyr Pong, Lukasz Kaiser, Jerry Tworek, Andrew Carr, Lilian Weng, Sandhini Agarwal, Karl Cobbe, Vineet Kosaraju, Alethea Power, Stanislas Polu, Jesse Han, Raul Puri, Shawn Jain, Benjamin Chess, Christian Gibson, Oleg Boiko, Emy Parparita, Amin Tootoonchian, Kyle Kosic, and Christopher Hesse. Introducing chatgpt, 2022. URL https://openai.com/blog/chatgpt.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. Green AI. Commun. ACM, 63(12): 54–63, 2020. doi: 10.1145/3381831. URL https://doi.org/10.1145/3381831.

Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In Regina Barzilay and Min-Yen Kan, editors, Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers, pages 1073–1083. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1099. URL https://doi.org/10.18653/v1/P17-1099.

Darsh J. Shah, Lili Yu, Tao Lei, and Regina Barzilay. Nutri-bullets: Summarizing health studies by composing segments. In Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021, pages 13780–13788. AAAI Press, 2021. doi: 10.1609/AAAI.V35I15.17624. URL https://doi.org/10.1609/aaai.v35i15.17624.

Or Sharir, Barak Peleg, and Yoav Shoham. The cost of training NLP models: A concise overview. CoRR, abs/2004.08900, 2020. URL https://arxiv.org/abs/2004.08900.

Eva Sharma, Chen Li, and Lu Wang. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, Proceedings of the

57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28-August 2, 2019, Volume 1: Long Papers, pages 2204–2213. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1212. URL https://doi.org/10.18653/v1/p19-1212.

Grishma Sharma and Deepak Sharma. Automatic text summarization methods: A comprehensive review. SN Comput. Sci., 4(1):33, 2023. doi: 10.1007/s42979-022-01446-w. URL https://doi.org/10.1007/s42979-022-01446-w.

Zejiang Shen, Kyle Lo, Lauren Yu, Nathan Dahlberg, Margo Schlanger, and Doug Downey. Multi-lexsum: Real-world summaries of civil rights lawsuits at multiple granularities. In NeurIPS, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/552ef803bef9368c29e53c167de34b55-Abstract-Datasets_and_Benchmarks.html.

Alex Sherstinsky. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. CoRR, abs/1808.03314, 2018. URL http://arxiv.org/abs/1808.03314.

Abhay Shukla, Paheli Bhattacharya, Soham Poddar, Rajdeep Mukherjee, Kripabandhu Ghosh, Pawan Goyal, and Saptarshi Ghosh. Legal case document summarization: Extractive and abstractive methods and their evaluation. In Yulan He, Heng Ji, Yang Liu, Sujian Li, Chia-Hui Chang, Soujanya Poria, Chenghua Lin, Wray L. Buntine, Maria Liakata, Hanqi Yan, Zonghan Yan, Sebastian Ruder, Xiaojun Wan, Miguel Arana-Catania, Zhongyu Wei, Hen-Hsen Huang, Jheng-Long Wu, Min-Yuh Day, Pengfei Liu, and Ruifeng Xu, editors, Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2022 - Volume 1: Long Papers, Online Only, November 20-23, 2022, pages 1048–1064. Association for Computational Linguistics, 2022. URL https://aclanthology.org/2022.aacl-main.77.

Rohit Singh, Venkata Vamsikrishna Meduri, Ahmed K. Elmagarmid, Samuel Madden, Paolo Papotti, Jorge-Arnulfo Quiané-Ruiz, Armando Solar-Lezama, and Nan Tang. Generating concise entity matching rules. In Semih Salihoglu, Wenchao Zhou, Rada Chirkova, Jun Yang, and Dan Suciu, editors, Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017, Chicago, IL, USA, May 14-19, 2017, pages 1635–1638. ACM, 2017. doi: 10.1145/3035918.3058739. URL https://doi.org/10.1145/3035918.3058739.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, pages 2888–2913. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EMNLP-MAIN.230. URL https://doi.org/10.18653/v1/2021.emnlp-main.230.

Shamane Siriwardhana, Rivindu Weerasekera, Tharindu Kaluarachchi, Elliott Wen, Rajib Rana, and Suranga Nanayakkara. Improving the domain adaptation of retrieval augmented generation (RAG) models for open domain question answering. Trans. Assoc. Comput. Linguistics, 11:1–17, 2023. URL https://transacl.org/ojs/index.php/tacl/article/view/4029.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, Elton Zheng, Rewon Child, Reza Yazdani Aminabadi, Julie Bernauer, Xia Song, Mohammad Shoeybi, Yuxiong He, Michael Houston, Saurabh Tiwary, and Bryan Catanzaro. Using deepspeed and megatron to train megatron-turing NLG 530b, A large-scale generative language model. CoRR, abs/2201.11990, 2022. URL https://arxiv.org/abs/2201.11990.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/c3a690be93aa602ee2dc0ccab5b7b67e-Abstract.html.

Pengxiao Song et al. LaWGPT. https://github.com/pengxiao-song/LaWGPT/tree/main, 2023.

Francesco Sovrano, Monica Palmirani, Biagio Distefano, Salvatore Sapienza, and Fabio Vitali. A dataset for evaluating legal question answering on private international law. In Juliano Maranhão and Adam Zachary Wyner, editors, ICAIL '21: Eighteenth International Conference for Artificial Intelligence and Law, São Paulo Brazil, June 21 - 25, 2021, pages 230–234. ACM, 2021. doi: 10.1145/3462757.3466094. URL https://doi.org/10.1145/3462757.3466094.

Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In Satinder Singh and Shaul Markovitch, editors, Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA, pages 4444–4451. AAAI Press, 2017. doi: 10.1609/AAAI.V31I1.11164. URL https://doi.org/10.1609/aaai.v31i1.11164.

Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A. Alemi, and Andrew Gordon Wilson. Does knowledge distillation really work? In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pages 6906–6919, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/376c6b9ff3bedbbea56751a84fffc10c-Abstract.html.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for modern deep learning research. In The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 13693–13696. AAAI Press, 2020. doi: 10.1609/AAAI.V34I09.7123. URL https://doi.org/10.1609/aaai.v34i09.7123.

Dan Su, Yan Xu, Tiezheng Yu, Farhad Bin Siddique, Elham J. Barezi, and Pascale Fung. Caire-covid: A question answering and query-focused multi-document summarization system for COVID-19 scholarly information management. In Karin Verspoor, Kevin Bretonnel Cohen, Michael Conway, Berry de Bruijn, Mark Dredze, Rada Mihalcea, and Byron C. Wallace, editors, Proceedings of the 1st Workshop on NLP for COVID-19@ EMNLP 2020, Online, December 2020. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.NLPCOVID19-2.14. URL https://doi.org/10.18653/v1/2020.nlpcovid19-2.14.

Yixuan Su and Nigel Collier. Contrastive search is what you need for neural text generation. Trans. Mach. Learn. Res., 2023, 2023. URL https://openreview.net/forum?id=GbkWw3jwL9.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. A contrastive framework for neural text generation. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/871cae8f599cb8bbfcb0f58fe1af95ad-Abstract-Conference.html.

Octavia-Maria Sulea, Marcos Zampieri, Mihaela Vela, and Josef van Genabith. Predicting the law area and decisions of french supreme court cases. In Ruslan Mitkov and Galia Angelova, editors, Proceedings of the International Conference Recent Advances in Natural Language Processing,

RANLP 2017, Varna, Bulgaria, September 2 - 8, 2017, pages 716–722. INCOMA Ltd., 2017. doi: 10.26615/978-954-452-049-6_092. URL https://doi.org/10.26615/978-954-452-049-6_092.

Zhongxiang Sun. A short survey of viewing large language models in legal aspect. CoRR, abs/2303.09136, 2023. doi: 10.48550/ARXIV.2303.09136. URL https://doi.org/10.48550/arXiv.2303.09136.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, pages 3104–3112, 2014. URL https://proceedings.neurips.cc/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2-Abstract.html.

Ayesha Ayub Syed, Ford Lumban Gaol, and Tokuro Matsuo. A survey of the state-of-the-art models in neural abstractive text summarization. IEEE Access, 9:13248–13265, 2021. doi: 10.1109/ACCESS.2021.3052783. URL https://doi.org/10.1109/ACCESS.2021.3052783.

Andrea Tagarelli and Andrea Simeri. Unsupervised law article mining based on deep pre-trained language representation models with application to the italian civil code. Artif. Intell. Law, 30(3):417–473, 2022. doi: 10.1007/S10506-021-09301-8. URL https://doi.org/10.1007/s10506-021-09301-8.

Thierry Tambe, Coleman Hooper, Lillian Pentecost, Tianyu Jia, En-Yu Yang, Marco Donato, Victor Sanh, Paul N. Whatmough, Alexander M. Rush, David Brooks, and Gu-Yeon Wei. Edgebert: Sentence-level energy optimizations for latency-aware multi-task NLP inference. In MICRO '21: 54th Annual IEEE/ACM International Symposium on Microarchitecture, Virtual Event, Greece, October 18-22, 2021, pages 830–844. ACM, 2021. doi: 10.1145/3466752.3480095. URL https://doi.org/10.1145/3466752.3480095.

Alex Tamkin, Miles Brundage, Jack Clark, and Deep Ganguli. Understanding the capabilities, limitations, and societal impact of large language models. CoRR, abs/2102.02503, 2021. URL https://arxiv.org/abs/2102.02503.

Wang-Chiew Tan, Yuliang Li, Pedro Rodriguez, Richard James, Xi Victoria Lin, Alon Y. Halevy, and Wen-tau Yih. Reimagining retrieval augmented language models for answering queries. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023, pages 6131–6146. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-ACL.382. URL https://doi.org/10.18653/v1/2023.findings-acl.382.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. Multilingual translation from denoising pre-training. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021, volume ACL/IJCNLP 2021 of Findings of ACL, pages 3450–3466. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.FINDINGS-ACL.304. URL https://doi.org/10.18653/v1/2021.findings-acl.304.

Dhaval Taunk, Lakshya Khanna, Siri Venkata Pavan Kumar Kandru, Vasudeva Varma, Charu Sharma, and Makarand Tapaswi. Grapeqa: Graph augmentation and pruning to enhance question-answering. In Ying Ding, Jie Tang, Juan F. Sequeda, Lora Aroyo, Carlos Castillo, and Geert-Jan Houben, editors, Companion Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023, pages 1138–1144. ACM, 2023. doi: 10.1145/3543873.3587651. URL https://doi.org/10.1145/3543873.3587651.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. ACM Comput. Surv., 55(6):109:1–109:28, 2023. doi: 10.1145/3530811. URL https://doi.org/10.1145/3530811.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and verification. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers), pages 809–819. Association for Computational Linguistics, 2018. doi: 10.18653/v1/n18-1074. URL https://doi.org/10.18653/v1/n18-1074.

Ozan Tonguz, Yiwei Qin, Yimeng Gu, and Hyun Hannah Moon. Automating claim construction in patent applications: The cmumine dataset. In Nikolaos Aletras, Ion Androutsopoulos, Leslie Barrett, Catalina Goanta, and Daniel Preotiuc-Pietro, editors, Proceedings of the Natural Legal Language Processing Workshop 2021, NLLP@EMNLP 2021, Punta Cana, Dominican Republic, November 10, 2021, pages 205–209. Association for Computational Linguistics, 2021. URL https://aclanthology.org/2021.nllp-1.21.

S. M. Towhidul Islam Tonmoy, S. M. Mehedi Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. A comprehensive survey of hallucination mitigation techniques in large language models. CoRR, abs/2401.01313, 2024. doi: 10.48550/ARXIV.2401.01313. URL https://doi.org/10.48550/arXiv.2401.01313.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. CoRR, abs/2302.13971, 2023a. doi: 10.48550/ARXIV.2302.13971. URL https://doi.org/10.48550/arXiv.2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. CoRR, abs/2307.09288, 2023b. doi: 10.48550/ARXIV.2307.09288. URL https://doi.org/10.48550/arXiv.2307.09288.

Vu D. Tran, Minh Le Nguyen, and Ken Satoh. Automatic catchphrase extraction from legal case documents via scoring using deep neural networks. CoRR, abs/1809.05219, 2018. URL http://arxiv.org/abs/1809.05219.

Don Tuggener, Pius von Däniken, Thomas Peetz, and Mark Cieliebak. LEDGAR: A large-scale multi-label corpus for text classification of legal provisions in contracts. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, and Stelios Piperidis, editors, Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020, pages 1235–1241. European Language Resources Association, 2020. URL https://aclanthology.org/2020.lrec-1.155/.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. Zephyr: Direct distillation of LM alignment. CoRR, abs/2310.16944, 2023. doi: 10.48550/ARXIV.2310.16944. URL https://doi.org/10.48550/arXiv.2310.16944.

Nisansala Ubayawardena and Maduka Hansani. A study on the effectiveness of using google translate in legal translation: With special reference to selected legal documents of the registrar general's department. 1, 07 2023.

Stefanie Urchs, Jelena Mitrovic, and Michael Granitzer. Design and implementation of german legal decision corpora. In Ana Paula Rocha, Luc Steels, and H. Jaap van den Herik, editors, Proceedings of the 13th International Conference on Agents and Artificial Intelligence, ICAART 2021, Volume 2, Online Streaming, February 4-6, 2021, pages 515–521. SCITEPRESS, 2021. doi: 10.5220/0010187305150521. URL https://doi.org/10.5220/0010187305150521.

Liam van der Poel, Ryan Cotterell, and Clara Meister. Mutual information alleviates hallucinations in abstractive summarization. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, pages 5956–5965. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.399. URL https://doi.org/10.18653/v1/2022.emnlp-main.399.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Enzo Veltri, Gilbert Badaro, Mohammed Saeed, and Paolo Papotti. Data ambiguity profiling for the generation of training examples. In 39th IEEE International Conference on Data Engineering, ICDE 2023, Anaheim, CA, USA, April 3-7, 2023, pages 450–463. IEEE, 2023. doi: 10.1109/ICDE55515.2023.00041. URL https://doi.org/10.1109/ICDE55515.2023.00041.

Jesse Vig, Alexander R. Fabbri, Wojciech Kryscinski, Chien-Sheng Wu, and Wenhao Liu. Exploring neural models for query-focused summarization. In Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz, editors, Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022, pages 1455–1468. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.FINDINGS-NAACL.109. URL https://doi.org/10.18653/v1/2022.findings-naacl.109.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. Diverse beam search for improved description of complex scenes. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, pages 7371–7379. AAAI Press, 2018. doi: 10.1609/AAAI.V32I1.12340. URL https://doi.org/10.1609/aaai.v32i1.12340.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pages 2692–2700, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/29921001f2f04bd3baee84a12e98098f-Abstract.html.

Andrew Vold and Jack G. Conrad. Using transformers to improve answer retrieval for legal questions. In Juliano Maranhão and Adam Zachary Wyner, editors, ICAIL '21: Eighteenth International Conference for Artificial Intelligence and Law, São Paulo Brazil, June 21 - 25, 2021, pages 245–249. ACM, 2021. doi: 10.1145/3462757.3466102. URL https://doi.org/10.1145/3462757.3466102.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. Fact or fiction: Verifying scientific claims. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pages 7534–7550. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.EMNLP-MAIN.609. URL https://doi.org/10.18653/v1/2020.emnlp-main.609.

David Wadden, Kyle Lo, Lucy Lu Wang, Arman Cohan, Iz Beltagy, and Hannaneh Hajishirzi. Multivers: Improving scientific claim verification with weak supervision and full-document context. In Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz, editors, Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022, pages 61–76. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.FINDINGS-NAACL.6. URL https://doi.org/10.18653/v1/2022.findings-naacl.6.

Wiebke Wagner. Steven bird, ewan klein and edward loper: Natural language processing with python, analyzing text with the natural language toolkit - o'reilly media, beijing, 2009, ISBN 978-0-596-51649-9. Lang. Resour. Evaluation, 44(4):421–424, 2010. doi: 10.1007/S10579-010-9124-X. URL https://doi.org/10.1007/s10579-010-9124-x.

David Wan and Mohit Bansal. Factpegasus: Factuality-aware pre-training and fine-tuning for abstractive summarization. In Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz, editors, Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022, pages 1010–1028. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.NAACL-MAIN.74. URL https://doi.org/10.18653/v1/2022.naacl-main.74.

Xiaojun Wan and Jianwu Yang. Improved affinity graph based multi-document summarization. In Robert C. Moore, Jeff A. Bilmes, Jennifer Chu-Carroll, and Mark Sanderson, editors, Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 4-9, 2006, New York, New York, USA. The Association for Computational Linguistics, 2006. URL https://aclanthology.org/N06-2046/.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pages 3261–3275, 2019a. URL https://proceedings.neurips.cc/paper/2019/hash/4496bf24afe7fab6f046bf4923da8de6-Abstract.html.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019b. URL https://openreview.net/forum?id=rJ4km2R5t7.

Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. Heterogeneous graph neural networks for extractive document summarization. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pages 6209–6219. Association for Computational Linguistics, 2020a. doi: 10.18653/V1/2020.ACL-MAIN.553. URL https://doi.org/10.18653/v1/2020.acl-main.553.

Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Darrin Eide, Kathryn Funk, Rodney Kinney, Ziyang Liu, William Merrill, Paul Mooney, Dewey A. Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex D. Wade, Kuansan Wang, Chris Wilhelm, Boya Xie, Douglas Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. CORD-19: the covid-19 open research dataset. CoRR, abs/2004.10706, 2020b. URL https://arxiv.org/abs/2004.10706.

Nancy Xin Ru Wang, Diwakar Mahajan, Marina Danilevsky, and Sara Rosenthal. Semeval-2021 task 9: Fact verification and evidence finding for tabular data in scientific documents (SEM-TAB-FACTS). In Alexis Palmer, Nathan Schneider, Natalie Schluter, Guy Emerson, Aurélie Herbelot, and Xiaodan Zhu, editors, Proceedings of the 15th International Workshop on Semantic Evaluation, SemEval@ACL/IJCNLP 2021, Virtual Event / Bangkok, Thailand, August 5-6, 2021, pages 317–326. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.SEMEVAL-1.39. URL https://doi.org/10.18653/v1/2021.semeval-1.39.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020c. URL https://proceedings.neurips.cc/paper/2020/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Yiming Wang, Zhuosheng Zhang, and Rui Wang. Element-aware summarization with large language models: Expert-aligned evaluation and chain-of-thought method. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023, pages 8640–8665. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.482. URL https://doi.org/10.18653/v1/2023.acl-long.482.

Yu Wang, Gu-Yeon Wei, and David Brooks. Benchmarking tpu, gpu, and CPU platforms for deep learning. CoRR, abs/1907.10701, 2019c. URL http://arxiv.org/abs/1907.10701.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net, 2022a. URL https://openreview.net/forum?id=gEZrGCozdqR.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In NeurIPS, 2022b. URL http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Jerry W. Wei, Chengyu Huang, Soroush Vosoughi, and Jason Wei. What are people asking about covid-19? A question classification dataset. CoRR, abs/2005.12522, 2020. URL https://arxiv.org/abs/2005.12522.

Eva WIESMANN. Machine translation in the field of law: A study of the translation of italian legal texts into german. Comparative Legilinguistics, 37:117–153, Oct. 2019. doi: 10.14746/cl.2019.37.4. URL https://pressto.amu.edu.pl/index.php/cl/article/view/18273.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. CoRR, abs/1910.03771, 2019. URL http://arxiv.org/abs/1910.03771.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen, editors, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020, pages 38–45. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.EMNLP-DEMOS.6. URL https://doi.org/10.18653/v1/2020.emnlp-demos.6.

Dustin Wright, David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Isabelle Augenstein, and Lucy Lu Wang. Generating scientific claims for zero-shot scientific fact checking. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 2448–2460. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.175. URL https://doi.org/10.18653/v1/2022.acl-long.175.

Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga Behram, Jinshi Huang, Charles Bai, Michael Gschwind, Anurag Gupta, Myle Ott, Anastasia Melnikov, Salvatore Candido, David Brooks, Geeta Chauhan, Benjamin Lee, Hsien-Hsin S. Lee, Bugra Akyildiz, Maximilian Balandat, Joe Spisak, Ravi Jain, Mike Rabbat, and Kim M. Hazelwood. Sustainable AI: environmental implications, challenges and opportunities. In Diana Marculescu, Yuejie Chi, and Carole-Jean Wu, editors, Proceedings of Machine Learning and Systems 2022, MLSys 2022, Santa Clara, CA, USA, August 29 - September 1, 2022. mlsys.org, 2022. URL https://proceedings.mlsys.org/paper/2022/hash/ed3d2c21991e3bef5e069713af9fa6ca-Abstract.html.

Jeff Wu, Long Ouyang, Daniel M. Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul F. Christiano. Recursively summarizing books with human feedback. CoRR, abs/2109.10862, 2021a. URL https://arxiv.org/abs/2109.10862.

Wenhao Wu, Wei Li, Xinyan Xiao, Jiachen Liu, Ziqiang Cao, Sujian Li, Hua Wu, and Haifeng Wang. BASS: boosting abstractive summarization with unified semantic graph. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, pages 6052–6067. Association for Computational Linguistics, 2021b. doi: 10.18653/V1/2021.ACL-LONG.472. URL https://doi.org/10.18653/v1/2021.acl-long.472.

Yuanhao Wu, Juno Zhu, Siliang Xu, Kashun Shum, Cheng Niu, Randy Zhong, Juntong Song, and Tong Zhang. Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. CoRR, abs/2401.00396, 2024. doi: 10.48550/ARXIV.2401.00396. URL https://doi.org/10.48550/arXiv.2401.00396.

Yuxiang Wu and Baotian Hu. Learning to extract coherent summary via deep reinforcement learning. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, pages 5602–5609. AAAI Press, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16838.

Wen Xiao and Giuseppe Carenini. Systematically exploring redundancy reduction in summarizing long documents. In Kam-Fai Wong, Kevin Knight, and Hua Wu, editors, Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2020, Suzhou, China, December 4-7, 2020, pages 516–528. Association for Computational Linguistics, 2020. URL https://aclanthology.org/2020.aacl-main.51/.

Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohan. PRIMERA: pyramid-based masked sentence pre-training for multi-document summarization. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 5245–5263. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.ACL-LONG.360. URL https://doi.org/10.18653/v1/2022.acl-long.360.

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021, pages 14138–14148. AAAI Press, 2021. doi: 10.1609/AAAI.V35I16.17664. URL https://doi.org/10.1609/aaai.v35i16.17664.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. Discourse-aware neural extractive text summarization. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pages 5021–5031. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.ACL-MAIN.451. URL https://doi.org/10.18653/v1/2020.acl-main.451.

Avaneesh Kumar Yadav, Ashish Kumar Maurya, Ranvijay, and Rama Shankar Yadav. Extractive text summarization using recent approaches: A survey. Ingénierie des Systèmes d Inf., 26(1):109–121, 2021. doi: 10.18280/ISI.260112. URL https://doi.org/10.18280/isi.260112.

Divakar Yadav, Rishabh Katna, Arun Kumar Yadav, and Jorge Morato. Feature based automatic text summarization methods: A comprehensive state-of-the-art survey. IEEE Access, 10:133981–134003, 2022. doi: 10.1109/ACCESS.2022.3231016. URL https://doi.org/10.1109/ACCESS.2022.3231016.

Min Yang, Qiang Qu, Wenting Tu, Ying Shen, Zhou Zhao, and Xiaojun Chen. Exploring human-like reading strategy for abstractive text summarization. In The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, pages 7362–7369. AAAI Press, 2019. doi: 10.1609/AAAI.V33I01.33017362. URL https://doi.org/10.1609/aaai.v33i01.33017362.

Tien-Ju Yang, Yu-Hsin Chen, Joel S. Emer, and Vivienne Sze. A method to estimate the energy consumption of deep neural networks. In Michael B. Matthews, editor, 51st Asilomar Conference on Signals, Systems, and Computers, ACSSC 2017, Pacific Grove, CA, USA, October 29 - November 1, 2017, pages 1916–1920. IEEE, 2017. doi: 10.1109/ACSSC.2017.8335698. URL https://doi.org/10.1109/ACSSC.2017.8335698.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. QA-GNN: reasoning with language models and knowledge graphs for question answering. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021, pages 535–546. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.NAACL-MAIN.45. URL https://doi.org/10.18653/v1/2021.naacl-main.45.

Michihiro Yasunaga, Antoine Bosselut, Hongyu Ren, Xikun Zhang, Christopher D. Manning, Percy Liang, and Jure Leskovec. Deep bidirectional language-knowledge graph pretraining. In NeurIPS, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/f224f056694bcfe465c5d84579785761-Abstract-Conference.html.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. Zerogen: Efficient zero-shot learning via dataset generation. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, pages 11653–11669. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.801. URL https://doi.org/10.18653/v1/2022.emnlp-main.801.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woo-Myoung Park. Gpt3mix: Leveraging large-scale language models for text augmentation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021, pages 2225–2239. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.FINDINGS-EMNLP.192. URL https://doi.org/10.18653/v1/2021.findings-emnlp.192.

Tiezheng Yu, Zihan Liu, and Pascale Fung. Adaptsum: Towards low-resource domain adaptation for abstractive summarization. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021, pages 5892–5904. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.NAACL-MAIN.471. URL https://doi.org/10.18653/v1/2021.naacl-main.471.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. Large language model as attributed training data generator: A tale of diversity and bias. CoRR, abs/2306.15895, 2023. doi: 10.48550/ARXIV.2306.15895. URL https://doi.org/10.48550/arXiv.2306.15895.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. Bartscore: Evaluating generated text as text generation. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pages 27263–27277, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/e4d2b6e6fdeca3e60e0f1a62fee3d9dd-Abstract.html.

Salud María Jiménez Zafra, Roser Morante, María Teresa Martín-Valdivia, and Luis Alfonso Ureña López. Corpora annotated with negation: An overview. Comput. Linguistics, 46(1):1–52, 2020. doi: 10.1162/COLI_A_00371. URL https://doi.org/10.1162/coli_a_00371.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/c8512d142a2d849725f31a9a7a361ab9-Abstract.html.

Sina Zarrieß, Henrik Voigt, and Simeon Schüz. Decoding methods in neural language generation: A survey. Inf., 12(9):355, 2021. doi: 10.3390/INFO12090355. URL https://doi.org/10.3390/info12090355.

Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Marcin Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael S. Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Pete Florence. Socratic models: Composing zero-shot multimodal reasoning with language. In The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net, 2023. URL https://openreview.net/pdf?id=G2Q2Mh3avow.

Haoyu Zhang, Jingjing Cai, Jianjun Xu, and Ji Wang. Pretraining-based natural language generation for text summarization. In Mohit Bansal and Aline Villavicencio, editors, Proceedings of the 23rd Conference on Computational Natural Language Learning, CoNLL 2019, Hong Kong, China, November 3-4, 2019, pages 789–797. Association for Computational Linguistics, 2019. doi: 10.18653/v1/K19-1074. URL https://doi.org/10.18653/v1/K19-1074.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, volume 119 of Proceedings of Machine Learning Research, pages 11328–11339. PMLR, 2020a. URL http://proceedings.mlr.press/v119/zhang20ae.html.

Mengli Zhang, Gang Zhou, Wanting Yu, Ningbo Huang, and Wenfen Liu. A comprehensive survey of abstractive text summarization based on deep learning. Computational Intelligence and Neuroscience, 2022, 2022a.

Miao Zhang, Rufeng Dai, Ming Dong, and Tingting He. DRLK: dynamic hierarchical reasoning with language model and knowledge graph for question answering. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, pages 5123–5133. Association for Computational Linguistics, 2022b. doi: 10.18653/V1/2022.EMNLP-MAIN.342. URL https://doi.org/10.18653/v1/2022.emnlp-main.342.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020b. URL https://openreview.net/forum?id=SkeHuCVFDr.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D. Manning, and Jure Leskovec. Greaselm: Graph reasoning enhanced language models for question answering. CoRR, abs/2201.08860, 2022c. URL https://arxiv.org/abs/2201.08860.

Xinliang Frederick Zhang, Heming Sun, Xiang Yue, Simon M. Lin, and Huan Sun. COUGH: A challenge dataset and models for COVID-19 FAQ retrieval. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, pages 3759–3769. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EMNLP-MAIN.305. URL https://doi.org/10.18653/v1/2021.emnlp-main.305.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. Siren's song in the AI ocean: A survey on hallucination in large language models. CoRR, abs/2309.01219, 2023. doi: 10.48550/ARXIV.2309.01219. URL https://doi.org/10.48550/arXiv.2309.01219.

Yusen Zhang, Ansong Ni, Ziming Mao, Chen Henry Wu, Chenguang Zhu, Budhaditya Deb, Ahmed Hassan Awadallah, Dragomir R. Radev, and Rui Zhang. Summ$ˆn$: A multi-stage summarization framework for long input dialogues and documents. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 1592–1604. Association for Computational Linguistics, 2022d. doi: 10.18653/V1/2022.ACL-LONG.112. URL https://doi.org/10.18653/v1/2022.acl-long.112.

Zhisong Zhang, Emma Strubell, and Eduard H. Hovy. A survey of active learning for natural language processing. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi,

United Arab Emirates, December 7-11, 2022, pages 6166–6190. Association for Computational Linguistics, 2022e. doi: 10.18653/V1/2022.EMNLP-MAIN.414. URL https://doi.org/10.18653/v1/2022.emnlp-main.414.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models. CoRR, abs/2303.18223, 2023. doi: 10.48550/ARXIV.2303.18223. URL https://doi.org/10.48550/arXiv.2303.18223.

Lucia Zheng, Neel Guha, Brandon R. Anderson, Peter Henderson, and Daniel E. Ho. When does pretraining help?: assessing self-supervised learning for law and the casehold dataset of 53, 000+ legal holdings. In Juliano Maranhão and Adam Zachary Wyner, editors, ICAIL '21: Eighteenth International Conference for Artificial Intelligence and Law, São Paulo Brazil, June 21 - 25, 2021, pages 159–168. ACM, 2021. doi: 10.1145/3462757.3466088. URL https://doi.org/10.1145/3462757.3466088.

Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. Searching for effective neural extractive summarization: What works and what's next. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pages 1049–1058. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1100. URL https://doi.org/10.18653/v1/p19-1100.

Jianing Zhou and Suma Bhat. Paraphrase generation: A survey of the state of the art. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, pages 5075–5086. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EMNLP-MAIN.414. URL https://doi.org/10.18653/v1/2021.emnlp-main.414.

Tianxun Zhou and Keng-Hwee Chiam. Synthetic data generation method for data-free knowledge distillation in regression neural networks. Expert Syst. Appl., 227:120327, 2023. doi: 10.1016/J.ESWA.2023.120327. URL https://doi.org/10.1016/j.eswa.2023.120327.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. Universalner: Targeted distillation from large language models for open named entity recognition. CoRR, abs/2308.03279, 2023a. doi: 10.48550/ARXIV.2308.03279. URL https://doi.org/10.48550/arXiv.2308.03279.

Xiyou Zhou, Zhiyu Chen, Xiaoyong Jin, and William Yang Wang. HULK: an energy efficiency benchmark platform for responsible natural language processing. In Dimitra Gkatzia and Djamé Seddah, editors, Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, EACL 2021, Online, April 19-23, 2021, pages 329–336. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EACL-DEMOS.39. URL https://doi.org/10.18653/v1/2021.eacl-demos.39.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net, 2023b. URL https://openreview.net/pdf?id=92gvk82DE-.

Chenguang Zhu, Ziyi Yang, Robert Gmyr, Michael Zeng, and Xuedong Huang. Leveraging lead bias for zero-shot abstractive news summarization. In Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai, editors, SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July

11-15, 2021, pages 1462–1471. ACM, 2021. doi: 10.1145/3404835.3462846. URL https://doi.org/10.1145/3404835.3462846.

Junnan Zhu, Qian Wang, Yining Wang, Yu Zhou, Jiajun Zhang, Shaonan Wang, and Chengqing Zong. NCLS: neural cross-lingual summarization. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pages 3052–3062. Association for Computational Linguistics, 2019. doi: 10.18653/V1/D19-1302. URL https://doi.org/10.18653/v1/D19-1302.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji-Rong Wen. Large language models for information retrieval: A survey. CoRR, abs/2308.07107, 2023. doi: 10.48550/ARXIV.2308.07107. URL https://doi.org/10.48550/arXiv.2308.07107.