



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DOTTORATO DI RICERCA IN
COMPUTER SCIENCE AND ENGINEERING

Ciclo 38

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

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

OPTIMIZATION IN LARGE LANGUAGE MODELS: INNOVATIVE APPROACHES
LEVERAGING KNOWLEDGE DISTILLATION AND PRUNING

Presentata da: Paolo Italiani

Coordinatore Dottorato

Paola Salomoni

Supervisore

Gianluca Moro

Co-supervisore

Antonella Carbonaro

Esame finale anno 2026

Contents

Abstract	ix
1 Introduction	1
1.1 Natural Language Processing	2
1.1.1 Transformers and Pretrained Language Models	3
1.1.2 Transformer Models and Their Computational Complexity	3
1.1.3 Thesis Organization	4
2 Token Pruning	6
2.1 Model Compression Techniques for Transformers	6
2.2 Definition and Taxonomy of Token Pruning	7
2.3 Challenges in Token Pruning	9
2.4 Token Pruning Methodologies	11
2.4.1 Heuristic-Based Token Pruning Methods	11
2.4.2 Reinforcement Learning-Based Approaches	12
2.4.3 Differentiable and Learnable Pruning Methods	13
2.4.4 Prompt-Level Compression for LLMs	14
2.4.5 Hybrid and Multi-Stage Methods	16
2.5 Read Between the Tokens: Differentiable Text Pruning via Perturbed Top- k Selection	17
2.5.1 Introduction	18
2.5.2 Related Work	20
2.5.3 METHOD	22
2.5.4 Experimental Setup	25
2.5.5 Discussion	30
2.5.6 Conclusion	35
3 Knowledge Distillation	36
3.1 Definition and Taxonomy of Knowledge Distillation	36
3.2 Challenges in Knowledge Distillation	38
3.3 Knowledge Distillation Methodologies	42
3.3.1 Response-Based Distillation	42
3.3.2 Feature-Based Distillation	43

3.3.3	Relation-Based and Similarity-Based Distillation	44
3.3.4	Self-Distillation and Online Distillation Techniques	45
3.3.5	Multi-Teacher and Ensemble Distillation	46
3.3.6	Adversarial and Other Advanced Distillation Techniques	47
3.3.7	Data-Free Knowledge Distillation and Synthetic Data Generation	48
3.4	ACE-ATTORNEY: Enhancing Legal Question Answering with Data Generation and Knowledge Distillation from Large Language Models	49
3.4.1	Introduction	50
3.4.2	Related Work	52
3.4.3	ACE-ATTORNEY	53
3.4.4	Experimental Setup	57
3.4.5	Results and Discussion	61
3.4.6	Conclusion	67
4	Optimization through Minimal Architectural Modifications in Large Language Models	68
4.1	Introduction	68
4.2	Cogito Ergo <i>Summ</i> : Abstractive Summarization of Biomedical Papers via Semantic Parsing Graphs and Consistency Rewards	69
4.2.1	Introduction	69
4.2.2	Related Work	71
4.2.3	COGITOERGOSUMM Framework	73
4.2.4	Experimental Setup	78
4.2.5	Results and Ablation Studies	81
4.2.6	Conclusion	82
4.3	Evidence, my Dear Watson: Abstractive Dialogue Summarization on Learnable Relevant Utterances	82
4.3.1	Introduction	83
4.3.2	Related Work	85
4.3.3	Preliminaries	86
4.3.4	Method	87
4.3.5	Experimental Setup	90
4.3.6	Results	92
4.3.7	Conclusion	100
4.3.8	Reproducibility	101
4.3.9	ChatGPT Evaluation Details	104
4.3.10	Human Evaluation Details	105
4.3.11	Scientific Artifacts	105
4.4	MEMEWEAVER: Inter-Meme Graph Reasoning for Sexism and Misogyny Detection	106
4.4.1	Introduction	106
4.4.2	Related Work	108
4.4.3	MEMEWEAVER	109

4.4.4	Experimental Setup	112
4.4.5	Results & Discussion	116
4.4.6	Conclusion	118
4.4.7	Backbone Variant	120
4.4.8	Zero-shot LLM prompts	120
5	Conclusions	122
	Bibliography	124

List of Figures

2.1	Overview of our PRUNEPERT model. In this work, “Output” can refer to either a summary or a class.	19
2.2	PRUNEPERT architecture. In this example (for encoder–decoder models), the number of selected tokens is $k = 3$ and the number of perturbed copies is $u = 4$	23
2.3	Average execution time (seconds) per instance during pre-processing (Pre-Proc), forward (fw), and backward (bw) passes for summarization on SCILAY and classification with MISTRAL-7B. Peg-PP: Pegasus-PrunePert; LLML: LLMingua.	31
2.4	Human evaluations results considering faithfulness (Faith.), precision (Prec.), and recall (Rec.).	33
2.5	Average occurrence rates in order: nouns, punctuation, adjectives, verbs, determiners, other categories, and stopwords.	33
2.6	Differentiable top- k hyperparameters impact on model performance on BOOLQ classification dataset with MISTRAL-7B.	33
2.7	Normalized position distribution of the selected tokens from PRUNEPERT on the summarization and classification tasks.	34
3.1	ACE-ATTORNEY overview. (1) A frozen LLM generates {context, question, answer} triplets from short legal contexts. For each question, we retrieve the corresponding context, manually adjusting the answer if the context is incorrect. (2) We then train an SLM on these triplets. (3) During inference, we employ a Selective Generative Paradigm, where the SLM provides an answer based on the relevance of the retrieved document.	51
3.2	Human analysis on SYN-LEQA’s data quality according to the average <i>fluency</i> , <i>factualness</i> , and <i>informativeness</i> scores.	54
3.3	<i>Left</i> : HuggingFace checkpoints of the models used in this study. <i>Right</i> : Hyperparameters used for SLMs fine-tuning and inference, and zero-shot LLMs. † and ‡ are specific for SLMs and LLMs, respectively.	59
3.4	Graphical overview of model effectiveness, where dotted lines represent LLMs and solid lines SLMs.	64
3.5	Performance–efficiency relation. Top: SYN-LEQA; bottom: PRIVACYQA _{gen} . The size of bubbles refers to the number of model parameters.	65

3.6	Few-shot setting results for SYN-LEQA (SLQA) and PRIVACYQA _{gen} (PQA). The dashed black bands mark the top-performing LLM (STARLING-7B). .	66
3.7	Transferability results on SYN-LEQA. TRANSFERED denotes a model trained on one subset and tested on the other. ORIGINAL denotes a model trained and tested on the same subset.	66
4.1	Expressiveness comparison between binary relation extraction (OpenIE 5.1), EE, and AMR on two example biomedical sentences. <i>Italic</i> denotes node types.	72
4.2	Qualitative example of induced semantic parsing graphs and their assistance to high-quality summarization. Red text indicates hallucinated facts. Token background highlights node alignments.	74
4.3	Illustration of the COGITOERGOsumm architecture. We dig out semantic parsing graphs from the input text by means of event extraction and AMR. Each source document is fed to a transformer encoder; event and AMR graph embeddings are learned by edge-aware graph attention networks. A multi-granularity decoder then generates summaries based on all information levels: text, events, and abstract meaning representations. Document-summary semantic graph alignments are used to optimize model consistency via reinforcement learning.	75
4.4	Human evaluation scores on Informativeness , Factualness , Fluency , Succinctness , compared to the ROUGE-1/2/L average ($\tilde{R}$), BERTScore (BS), FactCC %, and average % novel n -grams w/ $n \in [1 - 4]$ (NN). COGITOERGOsumm (ours) achieves significantly higher ratings than BART-base PubMed (student t-test, $p=0.0305$).	81
4.5	Overview of DEARWATSON. Utterance annotation and summarization are simultaneously mastered.	85
4.6	Illustration of our joint and end-to-end DEARWATSON architectures. Following a shared initial phase for utterance classification, joint learning peculiarities are highlighted in red, while the alternative end-to-end modifications are shown in green. The ”//” symbol connotes backpropagation interruption points.	88
4.7	Relation between overall summarization ROUGE scores ($\mathcal{R}$) and the number of speakers/utterances in conversations.	95
4.8	Attention matrix visualization, the more intense the color, the higher the weight. A comparison between BART(DW_{JL}) (left) and BART (right). The vertical axis is the generated summary, and the horizontal axis is the source dialogue. Presented tokens are partial for ease of perception. Our framework guides the model to emphasize tokens that are more likely to be relevant for the summary.	97

4.9	Qualitative examples of predicted relevant utterances (highlighted in yellow) and their assistance to high-quality summarization and interpretability. Red text indicates generation errors. Taking BART as the backbone, we illustrate the inference of our fine-tuned joint learning model (left) and end-to-end model (right).	98
4.10	Runtime analysis (3-runs average) over the SAMSum and DialogSum test sets, BART-based backbone. Our model results include both annotation and generation times.	99
4.11	Human evaluation results. DEARWATSON joint learning and end-to-end learning achieve significantly higher recall and faithfulness than the previous state-of-the-art (student t-test, $p < 0.05$), corroborating the benefits of supervised input augmentation.	99
4.12	ChatGPT evaluation results. DEARWATSON end-to-end models exhibit higher precision and faithfulness.	100
4.13	Kendall-Tau correlation (%) of automatic metrics with human judgments.	101
4.14	ChatGPT annotation prompt (target: value on a 3-point Likert scale). [metric] equals "recall", "precision", or "faithfulness". The [input] can be a "resume" (recall/precision) or a "document" (faithfulness). [definition] spans explain the meaning of boundary min/max scores for the metric under investigation.	105
4.15	Human assessment interface.	105
4.16	Overview of MemeWeaver, our graph-based framework for end-to-end sexism/misogyny detection in memes. Motivated by the social dynamics of online hate, it models batches of memes (3 shown) as inter-connected nodes for more effective learning.	107
4.17	Architecture of MemeWeaver. Each meme undergoes text extraction (OCR, optionally enriched with LLM-generated captions) and is encoded by separate text and image encoders. The embeddings are fused, refined via inter-meme graph reasoning within each batch, and classified. Example shown with batch size $m = 3$.	109
4.18	Prompts used for LLM-based meme captioning. The LABEL placeholder was set to <i>misogynistic</i> or <i>sexist</i> depending on the downstream dataset.	110
4.19	Language distribution in EXIST. Ratio of sexist vs. non-sexist memes across splits.	113
4.20	Embedding space analysis. 2D visualizations of MAMI and EXIST test set embeddings $\mathbf{f}'$ and $\mathbf{f}$, generated with and without MEMEWEAVER's IMGR, respectively. Decision boundaries from a linear classifier show higher accuracy on $\mathbf{f}'$ (see scores below plots), underscoring the benefit of IMGR's representations.	115
4.21	Training loss evolution of MemeWeaver with and without IMGR across datasets. Loss is plotted over normalized global training steps.	116

4.22	F1 scores evolution as inference batch size increases for MAMI (top) and EXIST (bottom).	118
4.23	Affinity patterns in IMGR. Correlation between learned affinity scores and fused embedding similarity, with dataset-specific behaviors.	119
4.24	Zero-shot LLM prompts across datasets.	121

Abstract

Large Language Models (LLMs) exhibit impressive capabilities in crucial tasks such as natural language understanding, generation, and complex reasoning, holding the potential to significantly influence our society. Nevertheless, these capabilities are accompanied by substantial computational and memory demands, underscoring the pressing need to devise effective techniques that address their efficiency challenges. This thesis investigates a spectrum of optimization methodologies aimed at improving the efficiency and performance of LLMs. It first explores two principal paradigms, token pruning, which dynamically reduces redundant computation during inference, and knowledge distillation, which transfers knowledge from a larger teacher model to a smaller, more efficient student. Both techniques target efficiency, seeking to minimize resource consumption while maintaining competitive performance. Beyond these compression-oriented strategies, the thesis also examines a complementary dimension of optimization: architectural refinement. This line of inquiry focuses on enhancing model performance through minimal yet strategically chosen structural modifications. Rather than compressing or accelerating models, these methods aim to optimize their representational and inductive capacities via subtle, computationally economical adjustments, such as introducing gating mechanisms, reparameterizing attention, or altering normalization strategies. The unifying principle is that small architectural interventions can yield measurable improvements in generalization, robustness, and accuracy. Collectively, these approaches illuminate a broader understanding of LLM optimization—one that spans both efficiency-driven compression and performance-driven architectural enhancement—demonstrating that thoughtful design at multiple levels of abstraction can lead to more capable and resource-conscious language models.

Keywords – artificial intelligence, natural language processing, large language models, knowledge distillation, token pruning, model optimization, architectural refinement, efficiency, generalization, robustness.

Acknowledgements

I thank Maggioli S.p.A. for co-funding the scholarship that allowed me to fulfill my PhD. I also wish to thank Prof. Gianluca Moro (DISI, University of Bologna) for his supervision and scientific support essential to conducting my research. I also want to thank all my Ph.D. colleagues with whom I shared this journey.

Chapter 1

Introduction

Over the last few years, large language models (LLMs) have redefined the landscape of natural language processing (NLP), achieving unprecedented performance across a wide spectrum of tasks including question answering, summarization, reasoning, and dialogue generation. These models, built upon the transformer architecture, have demonstrated remarkable generalization and linguistic fluency, bringing us closer to the goal of machines that can understand and generate human language with human-like sophistication. Yet, the extraordinary success of LLMs comes at a considerable cost: their training and inference require substantial computational resources, high memory consumption, and large-scale data. These demands pose critical challenges for sustainability, scalability, and deployment in real-world environments.

This thesis addresses these challenges by exploring strategies for optimizing LLMs across three complementary dimensions: efficiency, architecture, and knowledge transfer. Specifically, it investigates how selective computation, architectural refinement, and distillation can lead to models that are both computationally lighter and more semantically powerful. The central hypothesis guiding this research is that efficiency and performance need not be mutually exclusive; rather, by carefully designing mechanisms that prioritize relevance, structure, and interpretability, it is possible to achieve language models that are simultaneously faster, smaller, and more accurate.

The first line of research focuses on efficiency through dynamic token selection. We introduce PRUNEPERT, a method that augments pretrained language models with a pruning layer to identify and retain only the most salient tokens during inference. This approach substantially reduces computation while maintaining, and in some cases improving, model accuracy. In parallel, we investigate knowledge distillation by introducing the ACE-ATTORNEY framework. In particular, we demonstrate how LLMs can serve as teachers to generate synthetic training data (SYN-LEQA) and train smaller, domain-specialized student models. These students not only rival their teachers in performance but also achieve significant reductions in runtime, parameter count, and carbon footprint.

Beyond token pruning and distillation the thesis explores optimization through architectural refinements, starting with semantic augmentation. In particular, we pro-

pose a framework for biomedical summarization that integrates domain-specific semantic graphs—such as Abstract Meaning Representations (AMRs) and event structures—directly into transformer decoders. By aligning linguistic content with formal semantics, our models achieve greater factual accuracy and informativeness without increasing computational overhead. This line of inquiry illustrates how structural and representational enhancements can complement compression techniques, emphasizing that performance gains can also emerge from better design rather than larger scale. Additionally, we develop DEARWATSON, an annotate-then-generate framework for abstractive dialogue summarization that explicitly models utterance relevance before generation. DEARWATSON achieves new state-of-the-art performance and enhances interpretability by allowing the model to justify its outputs through relevance markers. The final contributions of this thesis investigate multimodal reasoning as pathways to sustainable and socially aware NLP. More precisely, MEMEWEAVER introduces a graph-based reasoning approach for detecting sexism and misogyny in memes, combining visual and textual cues into a unified learning structure. These models underscore the versatility of efficient architectures and highlight the potential for applying optimization principles to socially impactful applications.

Collectively, these contributions advance a unified vision of LLM optimization—one that bridges compression, architecture, and interpretability. They demonstrate that intelligent design choices at multiple levels of abstraction can yield models that are not only computationally efficient but also semantically robust and transparent. The remainder of this chapter introduces the theoretical and methodological background underpinning this work, beginning with an overview of natural language processing and the transformer paradigm.

1.1 Natural Language Processing

Natural Language Processing (NLP) stands as a central pillar of Artificial Intelligence (AI), focusing on enabling machines to understand, interpret, and generate human language in a meaningful way. Positioned at the intersection of computer science, computational linguistics, and machine learning, NLP seeks to bridge the gap between human communication and computational logic [Min et al. \[2024\]](#).

Over the past decade, the field has witnessed a remarkable transformation driven by deep learning methodologies that have redefined the boundaries of machine understanding. From early architectures such as recurrent neural networks (RNNs) [Sherstinsky \[2018\]](#) and convolutional neural networks (CNNs) [Gu et al. \[2018\]](#) to the emergence of transformer-based frameworks [Vaswani et al. \[2017a\]](#), the evolution of NLP has been characterized by both architectural innovation and scalability. These models have enabled systems to capture intricate semantic dependencies and contextual relationships, yielding near-human performance in tasks such as sentiment analysis, question answering, text summarization, and translation [Zhou et al. \[2023a\]](#).

Parallel to algorithmic advancements, the rapid improvement in computational hardware—particularly GPUs and TPUs—has catalyzed the training of increasingly complex

and capable language models [Ignat et al. \[2023\]](#). This synergy between hardware and model design has led to an era of unprecedented progress in language generation systems, where the line between machine-produced and human-produced text continues to blur. Such advances have proven especially influential in tasks like text summarization, offering new frameworks for balancing efficiency, coherence, and informativeness in generated content.

1.1.1 Transformers and Pretrained Language Models

The transformer architecture represents a paradigm shift in NLP, fundamentally altering how models process and generate language. Introduced by [Vaswani et al. \[2017a\]](#), the transformer’s attention-centric design replaced sequential computation with parallelizable mechanisms that better capture global dependencies. Variants such as encoder-only BERT [Devlin et al. \[2019b\]](#) and decoder-only GPT [Radford et al. \[2019b\]](#) demonstrate the versatility of this architecture, each optimized for distinct tasks and training paradigms.

Within text summarization, transformers facilitate a more profound understanding of linguistic structure and contextual flow. Unlike traditional extractive summarization, which merely selects representative sentences from source documents, transformer-based models enable *abstractive* summarization—synthesizing new sentences that concisely express the original meaning [Yadav and Shukla \[2021\]](#). This represents a major step toward genuinely generative models capable of mimicking the human summarization process, where the output reflects both semantic fidelity and linguistic fluency [Yang et al. \[2019\]](#).

Building on this foundation, pretrained language models (PLMs) have redefined the modern NLP landscape. By training on vast corpora through self-supervised objectives such as masked language modeling [Sinha et al. \[2021\]](#) or autoregressive prediction [Radford et al. \[2019b\]](#), PLMs acquire broad linguistic and contextual representations that can be efficiently fine-tuned for downstream applications [Qiu et al. \[2020\]](#). Architectures like BART [Lewis et al. \[2020a\]](#) and T5 [Raffel et al. \[2020\]](#) exemplify the encoder-decoder paradigm, combining comprehension and generation capabilities to achieve state-of-the-art results across summarization, translation, and dialogue generation tasks [Liu and Lapata \[2019a\]](#).

Interestingly, even decoder-only models—originally designed for autoregressive generation—have recently demonstrated strong performance in summarization and other comprehension tasks, further blurring the distinction between model types [Wang et al. \[2023a\]](#). This trend underscores an ongoing evolution toward general-purpose large language models (LLMs) that unify multiple NLP capabilities within a single, scalable architecture.

1.1.2 Transformer Models and Their Computational Complexity

Transformer-based models have revolutionized natural language processing by using self-attention to capture long-range dependencies. However, a well-known drawback is the quadratic time and memory complexity of the self-attention mechanism with respect to

sequence length. Given an input sequence of length N , a standard Transformer must compute attention weights for $N \times N$ token pairs at each layer, leading to $O(N^2)$ operations. This quadratic growth becomes a major bottleneck when processing long documents or deploying large models (e.g. BERT, GPT, T5) in real-time systems. In practice, many state-of-the-art large language models (LLMs) use contexts of hundreds or thousands of tokens, making inference latency and memory usage problematic for long inputs.

Importantly, not every token in a long sequence contributes equally to the model’s output [Pan et al. \[2024a\]](#). In fact, research has observed that many tokens become redundant after being processed by a few Transformer layers. In other words, some token representations are “encoded” or contextualized early, such that further attention to them yields diminishing returns. This insight suggests an opportunity: if one can identify and remove less informative tokens partway through the model, the overall computation can be reduced without significantly impacting accuracy. This idea – reducing sequence length on the fly – underpins token pruning strategies. By eliminating redundant tokens during inference, we can alleviate the quadratic cost of attention on long sequences and even potentially improve focus on the remaining relevant tokens. Token pruning thus directly targets the chief scalability issue of Transformers (their sequence length dependence) by dynamically shortening the effective input length.

1.1.3 Thesis Organization

This main core of thesis is organized into 4 chapters that progressively explore efficiency optimization techniques for large language models through token pruning, knowledge distillation, and architectural modifications.

Chapter 2 establishes the theoretical and methodological foundations of token pruning in transformer-based language models. We begin with model compression techniques for transformers, followed by a comprehensive definition and taxonomy of token pruning approaches. The chapter surveys existing token pruning methodologies, including heuristic-based methods, reinforcement learning-based approaches, differentiable and learnable pruning methods, and prompt-level compression techniques for LLMs. We then present PRUNEPERT which introduces top-k selection methods and differentiable text pruning via perturbation, providing both theoretical analysis and experimental validation.

Chapter 3 investigates knowledge distillation as a complementary efficiency technique. After defining the taxonomy of knowledge distillation, we examine the challenges inherent in distilling knowledge from large language models and review existing distillation methodologies, including response-based, feature-based, and relation-based approaches, as well as self-distillation and adversarial techniques. The chapter culminates in our work ACE-ATTORNEY which demonstrates how knowledge distillation can enhance legal question answering systems by transferring capabilities from large models to more efficient ones while maintaining performance.

Chapter 4 explores optimization through minimal architectural modifications to large language models. We present three distinct approaches: first, COGITOERGOSUMM introduces abstractive summarization of biomedical papers using semantic parsing graphs and consistency rewards. Second, DEARWATSON develops abstractive dialogue summa-

rization techniques for learnable relevance assessment. Finally, MEMEWEAVER addresses the specialized task of meme graph reasoning for detecting sexism and misogyny, demonstrating how targeted architectural enhancements can improve model performance on complex, multimodal tasks.

Chapter 6 concludes the thesis by synthesizing the key findings across token pruning, knowledge distillation, and architectural optimization. We discuss the broader implications of our work for the development of efficient large language models and outline promising directions for future research in model efficiency and optimization.

Chapter 2

Token Pruning

2.1 Model Compression Techniques for Transformers

Before focusing on token pruning specifically, it is useful to situate it among the broader landscape of Transformer model compression and acceleration techniques. A variety of approaches have been explored to make large Transformer models more efficient:

- **Weight Pruning:** Removing unnecessary parameters (individual weights or entire neurons) to sparsify the model. This can be unstructured (pruning arbitrary weights) or structured (pruning whole attention heads, feed-forward blocks, or layers). By zeroing out less important weights, weight pruning reduces model size and can yield speedups on specialized hardware, though unstructured sparsity often provides limited speedup on general accelerators.
- **Quantization:** Reducing the numerical precision of model parameters and computations. For example, converting 32-bit floating point weights to 8-bit or 4-bit integers cuts memory usage and can increase inference speed by using efficient low-bit matrix multiplication. Modern Transformer deployments frequently use quantization (including 8-bit or 4-bit weight quantization) to accelerate inference with minimal loss in accuracy.
- **Knowledge Distillation:** Training a smaller student model to imitate the outputs (or internal representations) of a large teacher model. Distillation leverages the teacher’s knowledge to guide the smaller model, often achieving performance close to the original model but with far fewer parameters. Notable examples include DistilBERT and MiniLM, which compress large pretrained models into lighter ones by matching logits or attention patterns.
- **Low-Rank Factorization:** Approximating the large weight matrices with lower-rank decompositions. Techniques like matrix factorization or singular value decomposition (SVD) reduce the number of parameters by exploiting low-rank structure, thereby cutting down computation in linear layers.

- **Sparse or Memory-Efficient Attention:** Modifying the Transformer architecture to avoid quadratic attention on all pairs of tokens. Examples include sparse attention models like Longformer and BigBird that restrict each token’s attention to a subset of tokens (local neighborhoods, global tokens, or random subsets) to achieve linear complexity. Similarly, Linformer and Reformer use projections or hashing to reduce the effective sequence length in attention calculation. These architecture changes trade off some modeling flexibility for significant improvements in scalability.

Each of the above techniques tackles efficiency from a different angle (parameter sparsity, numerical precision, etc.). Token pruning stands out by addressing sequence length directly – it dynamically shortens the input per instance during inference. In contrast to static model pruning or architecture redesign, token pruning adaptively removes tokens in response to the input content. This makes it a particularly powerful approach for speeding up inference in scenarios where some inputs contain a lot of irrelevant or redundant tokens (e.g. long documents with filler text, or conversations with off-topic turns). Token pruning can also be complementary to the other compression techniques. For example, one could first apply weight pruning or quantization to shrink a model, and then further apply token pruning at runtime for additional speedup. In the following, we focus on token pruning in depth: its definition, varieties, challenges, and the methods that have been developed to implement it.

2.2 Definition and Taxonomy of Token Pruning

Token pruning is a dynamic model compression strategy that selectively removes tokens from the input sequence as the model processes it, in order to reduce computation while maintaining performance [Jianhong \[2025\]](#). Unlike static architectural pruning, which would, say, remove entire layers or neurons and apply to all inputs permanently, token pruning operates at inference time and can make different decisions for each input. The goal is to identify tokens that are least important for the model’s prediction and prune them out, so that subsequent layers do not waste resources on them. By decreasing the sequence length on the fly, token pruning directly reduces the number of attention computations and feed-forward operations required in later layers.

Formally, we can imagine a token pruning module that, at certain points in the Transformer (often after each layer or after a group of layers), examines the sequence of token representations and drops a subset of them based on some criterion. The pruned tokens are effectively omitted from further computation (in hard pruning), or in some cases their information might be merged or retained in a reduced form (in soft pruning, discussed below). Token pruning is typically applied during inference, but it may be configured or learned during the training phase on a downstream task. There are several ways to categorize token pruning methods. One useful taxonomy is based on how tokens are pruned and how adaptive the process is.

- **Hard vs. soft pruning:** In hard token pruning, selected tokens are entirely removed from the sequence – they will not be attended to or processed in subsequent

layers. This yields direct computational savings, since those tokens contribute no further, but it risks losing information if a pruned token was actually important. In soft token pruning, by contrast, tokens are not completely discarded; instead, their representations might be down-weighted or merged with others [Kong et al. \[2022\]](#). For example, a pruned token’s vector could be added into a neighboring token’s vector, or a gate could simply reduce its influence to near-zero. Soft pruning retains a shadow of the token’s information, potentially mitigating accuracy loss at the cost of smaller computational savings, since the token is still processed in some form, albeit perhaps with reduced dimension or effect.

- **Static vs. adaptive pruning:** Static token pruning methods use the same pruning configuration for all inputs. For instance, a static approach might always prune every sequence down to a fixed length (e.g. keep the first 128 tokens of every input), or drop tokens with attention scores below a fixed threshold, regardless of the input content. In contrast, adaptive token pruning methods decide which tokens to prune based on the token’s estimated importance for that particular input instance [Mahmood et al. \[2021\]](#). Adaptive methods often employ learnable gating mechanisms, dynamic thresholds, or other data-dependent criteria to tailor the pruning decisions to each sequence. Almost all modern token pruning research focuses on adaptive approaches, since the optimal number of tokens to keep can vary greatly between inputs (a short sentence vs. a long paragraph) and contexts.
- **Compression combination:** Some approaches are hybrid, combining token pruning with other compression techniques. For example, a hybrid method might perform token pruning and knowledge distillation, training a smaller model on the outputs of a larger model that uses token pruning. Another hybrid strategy is to combine token pruning with structured weight pruning or quantization. An example of this is assigning different tokens different bitwidth precision: a recent approach by Yang et al. assigns each token either 8-bit, 4-bit, or “0-bit” (dropped entirely) based on its importance as measured by attention probabilities. This effectively merges token pruning with quantization: unimportant tokens are pruned (0-bit, no computation), while important tokens are kept at higher precision. Such hybrid schemes aim to maximize efficiency gains by attacking multiple bottlenecks (both sequence length and parameter cost) simultaneously.

These categories are not mutually exclusive. For instance, most hard pruning methods are also adaptive, and many modern techniques incorporate some learnable component even if they start from a heuristic. The key dimensions to note are whether a method prunes tokens completely or partially, and whether the pruning decisions are predetermined or learned on the fly.

2.3 Challenges in Token Pruning

While token pruning is a promising approach for accelerating Transformers, it introduces several challenges and trade-offs that researchers must address:

- **Accurate token importance estimation:** The core problem is deciding which tokens to prune without performing an expensive exhaustive search. Determining each token’s importance to the final prediction is non-trivial. Many early approaches used proxies like attention scores or simple statistics to rank token importance, but these can be noisy or unreliable indicators in some cases. More advanced methods learn a separate module or policy to estimate token importance, but that itself must be efficient. The challenge is to compute token saliency in a way that is both accurate and cheap. If importance is misestimated (pruning a crucial token or keeping many unimportant ones), the model’s performance can suffer.
- **Discrete decisions and differentiability:** Removing tokens is a discrete operation, a token is either kept or dropped. This discreteness makes it hard to integrate pruning decisions into the standard training process, which relies on backpropagation of gradients, which assume differentiable operations. A significant challenge has been how to train models to prune tokens. Many solutions resort to Reinforcement Learning (RL) or other gradient-free methods to learn a token selection policy. However, RL training is often slow and complex, requiring careful reward design and convergence of an additional policy network. Other approaches relax the discrete selection into a continuous approximation. For instance, using a soft mask or the Gumbel-Softmax trick to allow gradients to flow. While such differentiable relaxations enable end-to-end training, they often introduce their own issue.
- **Achieving Actual Sparsity:** Closely related to differentiability is the challenge of inducing actual runtime sparsity. Some methods use soft pruning during training, they learn to attenuate or mask tokens, but do not physically remove them until inference. For example, a model might learn continuous gates for tokens and during training simply down-weight certain token embeddings. If at test time those with gate < 0.5 are dropped, there is a train–test gap introduced: the model never experienced tokens being completely missing during training. This gap can lead to performance drops when the hard pruning is applied. Moreover, if a method never truly removes tokens during training, the computational benefits only materialize at inference; the training process itself remains as expensive as the original model. Ensuring that a pruning approach yields actual speedups (and not just theoretical sparsity) often requires special techniques or hardware support. Without careful engineering, the overhead of making pruning decisions (e.g. sorting tokens by importance every layer) can eat the gains obtained from processing fewer tokens.
- **Balancing efficiency and accuracy:** Any aggressive pruning of tokens risks

omitting information that could be important for the task, thus hurting model accuracy. One challenge is finding the right trade-off that gives significant speedup with minimal performance loss. In tasks requiring fine-grained understanding (e.g. reading comprehension, named entity recognition, machine translation), even tokens that seem minor could alter the prediction if removed [Xu et al. \[2022\]](#). Many token pruning methods indeed observe a drop in metrics when high pruning rates are used. The art is to prune just the “redundant” parts while retaining the essential context. Some solutions involve gradually pruning, e.g. removing a small percentage of tokens at multiple intermediate layers instead of a large percentage all at once, to allow the model to adjust and minimize damage. Others impose a cap on how much of the sequence can be pruned or use budget-aware schemes that allocate more tokens to harder examples. Ensuring that efficiency gains do not come at an unacceptable cost to accuracy is a central concern in this research.

- **Generalization and adaptability:** A token pruning strategy tuned for one setting might not generalize to others. For example, a method developed on classification tasks might struggle on generation tasks if it prunes tokens that are later needed for decoding. Domain shifts (e.g. from news text to scientific text) can also break heuristic importance metrics. The challenge is to make token pruning policies robust across tasks and domains. Adaptive methods that learn from data can adjust to new domains better than fixed heuristics, but they may still have biases. Recent research has noted that hybrid approaches (combining multiple criteria or having learnable components) tend to be more resilient to domain shifts. Nevertheless, evaluating token pruning methods on a wide range of tasks is important to ensure they don’t overfit to the idiosyncrasies of a particular dataset.
- **Implementation complexity:** Finally, deploying token pruning in real systems can be challenging. Unlike static pruned models (which are fixed and can be optimized once), token pruning requires modifying the model’s execution dynamically per input. Many deep learning frameworks assume a fixed computation graph and sequence length for efficiency. Pruning tokens means that the sequence length may shrink layer by layer, which is not a typical scenario for libraries. This may require custom operations or breaking batches into irregular sizes. There is overhead in doing so, and naive implementations might not actually speed up if they can’t leverage the GPU effectively. Specialized kernels or runtime support are often needed to reap the full benefits. For example, pruning 50% of tokens should ideally halve the compute, but if done without careful memory management, those savings might not be realized. Thus, integrating token pruning into transformer frameworks and hardware in an optimal way is an engineering challenge.

In summary, token pruning introduces a complex optimization problem: it must choose a subset of tokens that preserves the model’s output, do so efficiently and ideally differentiably, and integrate with training and inference pipelines. Over the past few years, a variety of methods have been proposed to tackle these challenges, as we discuss next.

2.4 Token Pruning Methodologies

A rich body of work has emerged exploring different methodologies for token pruning. These can be broadly categorized by the approach used to decide which tokens to prune. From simple heuristics to fully learned policies, as well as methods focusing specifically on prompt compression for LLMs and hybrid strategies. We review each category and highlight key representative works, including methods from both NLP and related domains (vision, multi-modal) where applicable.

2.4.1 Heuristic-Based Token Pruning Methods

Early token pruning techniques relied on predefined heuristics or rules to identify unimportant tokens. These methods do not add new trainable parameters; instead, they use some metric derived from the model’s internal states (often attention scores or token dynamics) to make pruning decisions at inference time. A straightforward heuristic is based on Transformer attention weights: since attention mechanisms compute how strongly each token attends to others, one can assume that tokens receiving consistently low attention are less important. PoWER-BERT Goyal et al. [2020] is a seminal example that uses attention-based metrics to drop tokens. In PoWER-BERT, the model progressively prunes a fraction of tokens at each layer based on their attention scores. This method, removes tokens whose absence least affects the next layers, thereby reducing overall computation in a layerwise manner. It achieves decent inference speedups with minimal fine-tuning, but since it uses a fixed pruning schedule across all inputs, it may under-prune easy examples or over-prune difficult ones.

Another heuristic method is Learned Token Pruning (LTP) Kim et al. [2022]. Despite the word "Learned" in its name, LTP initially derives a token importance score from attention like the heuristic methods, and then prunes tokens whose scores fall below a learned threshold at each layer. It automates the search for the optimal per-layer pruning percentages by introducing a differentiable thresholding mechanism during fine-tuning. Essentially, LTP still relies on attention values to rank tokens, but it learns the cutoff values for those attention scores that decide pruning, rather than using a manually set threshold. This allows it to adapt the pruning aggressiveness to the task and layer. LTP avoids expensive top- k sorting by using threshold masks, which makes it computationally cheap at inference. Experiments showed LTP can improve throughput by roughly $2\times$ with only a minor drop in accuracy (often $\leq 1\%$) on tasks like GLUE, outperforming earlier static methods.

A related rule-based approach is to use token entropy or representational change across layers. The intuition is that if a token’s hidden state does not change much from one layer to the next (low entropy or low variance), the model has already extracted what it needs from that token. Such a token could be safely removed or merged without affecting final predictions. Some methods monitor token embeddings over layers and drop those that become "inactive." This was explored, for example, in the context of efficient object detection with vision transformers, where tokens (image patches) that had little feature change were pruned early. Entropy-based pruning serves as another

simple criterion that does not require additional training, though it may be task-specific to determine a proper threshold.

A very simple heuristic baseline is fixed-length or position-based pruning. For instance, always truncating inputs to the first k tokens (or some mix of head/tail tokens) is a crude pruning strategy that ensures a fixed sequence length. Similarly, one might remove tokens like stop-words or punctuation across the board. While these methods can yield speedups and are trivial to implement, they are not adaptive and often significantly hurt accuracy for tasks requiring the pruned information. For example, truncating a document to 512 tokens might ruin a summarization task if the conclusion lies beyond that cutoff. Thus, fixed pruning is usually only used as a baseline to illustrate the need for more intelligent adaptive methods.

Heuristic methods have also been extended beyond NLP. In the vision domain, an analogue of token pruning is patch pruning for Vision Transformers (ViTs). For example, one approach prunes image patches by analyzing multi-head attention maps in ViTs. Patches that receive consistently low attention from all other patches can be dropped as they likely correspond to uninformative regions in the image. Such methods, like [Luo et al. \[2022b\]](#), apply the same attention-score threshold idea in each layer of a ViT to progressively reduce the number of patches processed. This demonstrates the generality of attention-based heuristics: whether in text or images, attention scores can signal token or patch importance. However, pure heuristic approaches (in both NLP and vision) are fundamentally limited by the fact that they are not learned from data – the criteria are hand-designed. As a result, they may not capture subtle notions of importance, and they often prune in a greedy manner without considering global sequence context or the end-task objective.

2.4.2 Reinforcement Learning-Based Approaches

To overcome the limitations of fixed heuristics, some researchers have posed token pruning as a sequential decision-making problem that can be solved with RL. In these methods, a policy is trained to dynamically choose which tokens to keep or drop, typically with a reward that encourages efficiency (fewer tokens) while preserving model accuracy. TR-BERT [Ye et al. \[2021b\]](#) is a prime example that uses an RL formulation. TR-BERT introduces a policy network that at each step (or each layer of the Transformer) decides whether each token should remain or be removed, treating it as a multi-step decision process. The policy is trained with reinforcement learning, where it receives a reward based on the final task performance and the computational cost saved. For instance, the reward might be high if the model’s accuracy remains high and the number of tokens is low, and penalized if accuracy drops too much or too many tokens are kept. Through many training episodes, the policy learns which tokens can be dropped with minimal impact. This approach was shown to find non-trivial pruning strategies that outperform simple heuristics, and it can adapt the pruning dynamically per input.

However, RL-based methods come with their own challenges. They require a separate training loop, often nested outside the normal model training, to train the policy, which can be time-consuming. Designing the reward function and ensuring stable convergence

of the RL training is also non-trivial. TR-BERT’s authors had to carefully balance the reward to avoid degenerate solutions (e.g., trivially keeping all tokens to maximize accuracy, or pruning too much to maximize speed reward). Moreover, the learned policy may not generalize outside the conditions it was trained on, and RL methods can be data-hungry.

Another RL-based approach in the literature is from the Length-Adaptive Transformer (LAT) framework [Kim and Cho \[2020\]](#), which predates some of these methods. Instead of using attention scores, LAT used a controller trained (with policy gradients or evolutionary strategies) to choose a pruning configuration for each input, i.e., how many tokens to keep at each layer. In effect, it learns to dynamically adjust the sequence length as a function of the input. While this shares the same motivation as TR-BERT, the formulation may differ in whether decisions are token-level or layer-level.

Reinforcement learning allows the model to explore combinations of token drops that a greedy heuristic might miss. For example, a heuristic might never drop a particular token type because its score is marginally above threshold, but an RL policy might learn that dropping it in conjunction with another token still yields a correct answer and more overall savings. Despite these advantages, RL approaches have fallen somewhat out of favor in recent years for token pruning, mainly due to their complexity and the rise of differentiable alternatives. They remain an important part of the historical development, demonstrating that learned policies can outperform static rules in token selection. In practice, the extra training overhead is a consideration: RL-based pruning might be more suitable when extremely high efficiency is required and one is willing to invest in complex training, whereas simpler methods might suffice for moderate speedups.

2.4.3 Differentiable and Learnable Pruning Methods

A major trend in recent work is to make token pruning differentiable, integrating the pruning decisions into the end-to-end training of the model. The goal is to allow gradient-based learning to directly optimize which tokens to prune, rather than relying on external criteria or separate RL loops. These methods introduce learnable modules (such as small classifiers or gating functions) into the Transformer that produce a score or decision for each token, indicating its importance. The challenge of non-differentiability is often addressed by using soft continuous relaxations of the pruning operation during training.

One line of work uses soft masking: each token is given a learnable continuous-valued gate g_i between 0 and 1, which multiplies the token’s representation. A value of $g_i = 0$ would effectively remove the token, while $g_i = 1$ keeps it. During training, the gates are treated as differentiable parameters, and a regularization term encourages many gates to drop to 0 promoting sparsity. At inference, any token whose gate falls below a threshold is pruned (hard decision). AdapLeR [Modarressi et al. \[2022b\]](#) follows this paradigm: it trains lightweight per-layer contribution predictors that output something akin to g_i for each token, and uses a soft-removal mechanism to gradually eliminate low-contributing tokens. At the start, all tokens are present; as the model trains, the contribution predictors learn which tokens can be given less weight. The soft removal is implemented in a way that the model can learn without a sudden drop of tokens, thus

avoiding a large shock to the network. By the end of training, many tokens have g_i near 0 and can be pruned out, yielding a speedup. AdapLeR showed that this learned gating approach can achieve better accuracy-speed trade-offs than static rules, because the model itself learns which tokens it truly needs for the task. It is fully differentiable (no RL needed), aside from the non-differentiable final thresholding of gates.

Another prominent method is Pyramid-BERT [Huang et al. \[2022a\]](#), which takes a slightly different approach grounded in core-set selection theory. Pyramid-BERT aims for a principled way to choose a subset of tokens that best represents the whole sequence (a “core-set”). It formulates token selection as a δ -covering problem and uses a k -Center greedy algorithm to pick tokens such that every token not chosen is within some radius δ in embedding space of a chosen token. This selection is done in a parallelizable manner with exponentially decaying number of tokens per layer (hence the “pyramid” – gradually shrinking the sequence length as layers go up). The important aspect is that Pyramid-BERT integrates this selection into end-to-end fine-tuning of BERT without needing auxiliary losses or RL. Essentially, at each layer it solves an optimization (greedy approximation of k -center) to drop tokens, and the model is fine-tuned knowing this will happen. This is not gradient-based in the usual sense (the selection step is a combinatorial algorithm), but it is deterministic and built into the training loop. By the end, the model has a fixed schedule of token counts per layer, significantly reducing complexity. Pyramid-BERT achieved solid results, showing that even without learning new parameters for pruning, one can use a theoretically grounded selection strategy to guide pruning in a training-aware way. One limitation is that the method is relatively complex and tailored to encoder models; it doesn’t straightforwardly extend to decoder or encoder-decoder models because it fixes a pyramid structure of pruning.

An influential development in differentiable pruning is the introduction of differentiable Top- k or sorting operators. Normally, selecting the top- k tokens by some score (like attention or a learned score) is non-differentiable. Some works have addressed this by creating smooth approximations to the top- k operation. [Petersen et al. \[2022\]](#) proposed a perturbation-based relaxation for selecting image patches (tokens). They added random noise to patch importance scores and used continuous approximations so that the selection of top patches became differentiable on average. This idea was later adapted to NLP token pruning by studies that wanted to avoid RL. However, a drawback observed was that while the selection could be made differentiable during training, the method did not enforce hard sparsity. In particular, many tokens would get fractional inclusion rather than being strictly dropped, so additional steps were needed to produce a final pruned model.

2.4.4 Prompt-Level Compression for LLMs

With the advent of very large language models and the focus on long-context tasks, a special category of token pruning has emerged, prompt or context compression for LLMs. In scenarios like few-shot prompting or long document QA, a large model might be given thousands of tokens of context (e.g. a long prompt with examples or a user’s conversation history). Processing all these tokens through every layer of the model is

extremely expensive. Prompt-level compression methods aim to pre-process or compress the input context by pruning or abstracting tokens before or outside the main LLM, so that the LLM receives a shorter, but information-rich, prompt.

One notable work is LLMingua [Jiang et al. \[2023c\]](#), which introduces a coarse-to-fine prompt compression pipeline tailored for generative LLMs. LLMingua uses a small auxiliary selector model to iteratively prune tokens from the prompt while trying to retain the key information needed by the big model. It consists of following components: (i) a perplexity-based budget controller that decides how many tokens can be removed while still keeping the prompt understandable to the LLM, (ii) an iterative token-level pruning mechanism that repeatedly removes the least important tokens (as measured by how much they increase perplexity or some loss if dropped), and (iii) a distribution alignment module that adjusts the smaller selector model’s judgments to account for the big LLM’s differing vocabulary or style. The distribution alignment is important because a small model might not value information the same way a huge LLM does, so they align the representations to avoid the small model pruning something that the big model would consider important. Using this framework, LLMingua was able to compress prompts to extreme degrees (keeping only, say, 10% of tokens) and still have the large model generate outputs that preserve the essential content. This was demonstrated in tasks like long text summarization and story generation. By selectively dropping tokens from the prompt, they achieved large speedups in LLM inference with minimal change in the generated text’s meaning.

An extension of this is LLMINGUA-2 [Pan et al. \[2024b\]](#), which further improved prompt compression by introducing a learnable reward model to guide the compression. Instead of just using perplexity and heuristic alignment, LLMingua-2 trains a reward function that explicitly scores how faithful a compressed prompt is to the original (for example, whether an answer generated with the compressed prompt would be the same as with the full prompt). This reward model, combined with a length budget, allows the system to adaptively decide which tokens to keep under strict length limits. Essentially, it casts prompt compression as a constrained optimization. In particular, maximize fidelity under a token budget, with a learned model predicting fidelity. This method achieved even higher fidelity summaries under compression, making it viable to compress prompts for very strict length constraints (e.g. compressing a 2000-token prompt to 200 tokens for a GPT-style model) with only minor loss in output quality. LLMingua-2 represents a convergence of token pruning with ideas from distillation and reinforcement learning.

Another approach targeting LLM context is ICAE [Ge et al. \[2024\]](#). ICAE takes an autoencoder perspective, it trains a small encoder (context compressor) that prunes or compresses the input by encoding it into a shorter sequence, and a corresponding decoder (corrector) module inside the large model that can reconstruct or utilize that compressed representation. In practice, ICAE’s encoder might drop tokens that it deems unimportant and output a compacted sequence of “content tokens,” and the large model is augmented with a decoder that can interpret those content tokens as if they were the full input. The corrector ensures that even if some tokens were pruned, the model can infer or restore missing details when needed. This two-stage framework (compress then

reconstruct) allows aggressive pruning without permanent information loss. Anything pruned can potentially be guessed by the large model’s corrector if it was predictable from context. ICAE showed strong results on tasks like long text classification and QA, achieving efficient inference by substantially shortening the context given to the LLM, while preserving task-critical information through the autoencoder mechanism.

Prompt-level compression techniques are essentially token pruning applied externally or at the input level of LLMs, often using a separate smaller model to do the pruning. This is attractive because it means one does not need to modify the large model’s architecture (which might be frozen, as in API-provided models like GPT-4). Instead, you have a smart preprocessing step that feeds a shorter prompt to the LLM. The small compressor model can often be trained cheaply or even be a fine-tuned version of a medium-sized model like GPT-2 or T5. These approaches have strong connections to classic techniques like summarization or sentence compression, but they optimize specifically for use with LLMs, maintaining not just gist but also the factual or task-specific details the LLM needs.

2.4.5 Hybrid and Multi-Stage Methods

Finally, some methods are hybrids that combine elements of the above categories or perform pruning in multiple stages. The motivation for hybrid approaches is to leverage the strengths of different methods and offset their weaknesses. For example, a hybrid approach might use a fast heuristic to do an initial coarse pruning and then a learned module for fine-grained decisions on the remaining tokens. This two-stage strategy can save computation (by quickly removing obviously unimportant tokens with a cheap rule) and then apply more expensive or accurate techniques on the borderline cases.

A form of hybridization can be combining token pruning with standard model pruning or distillation. For instance, one might first apply knowledge distillation to train a smaller Transformer, and also equip it with a token pruning mechanism, effectively compressing the model in two dimensions (parameters and sequence length). As referenced in [Jianhong \[2025\]](#), some works combine token and weight pruning to maximize speedup. Token pruning saves computation in self-attention (which depends on sequence length), while weight pruning saves compute in feed-forward layers and reduces memory. Together, they can substantially cut overall FLOPs. Empirical results suggest that doing both can be beneficial and that the two techniques don’t completely overlap in effect. The challenge in hybrid pruning is ensuring that the combined approach doesn’t excessively harm accuracy. For example, pruning weights might make the model more fragile to token removal, so it requires careful tuning or multi-task objectives to balance.

In some cases, hybrid refers to methods that integrate multiple criteria for pruning. Instead of relying on a single metric like attention or a single learned module, a hybrid method might, for example, prune tokens that are low-attention and have low gradient norms, or some combination of signals. The intuition is to be more robust, if a token is unimportant by several measures, we can be more confident in pruning it. While not many works explicitly advertise such multi-criterion pruning, researchers have explored related ideas like using both attention scores and MLM prediction confidence to decide

token importance. One concrete example of a multi-step hybrid approach is Leap-of-Though Kim et al. [2023], although primarily it addresses reasoning with truncated contexts, it references combining early exiting with token reduction. Early exit (layer-wise stopping of computation for easy inputs) can be combined with token pruning (reducing length for parts of the input), creating a joint policy where at any layer some tokens may finish processing (exit) while others continue. This kind of approach blurs the line between layer skipping and token skipping and is at the frontier of efficient inference research.

In summary, hybrid methods underscore that token pruning need not be a standalone technique, it can be part of a larger system of optimizations. By mixing heuristic and learned decisions, or merging token pruning with other compression methods, one can often achieve better efficiency without sacrificing as much accuracy. The trade-off, of course, is added complexity in design and tuning.

Moving forward, token pruning stands as an important piece in the puzzle of making huge models more efficient. It complements techniques like sparse attention (which handles very long sequences by design) by allowing models to adaptively concentrate compute on informative parts of the input. As LLMs continue to grow and be deployed, we expect token pruning and prompt compression techniques to become increasingly relevant. They provide a form of conditional computation, the model uses full capacity only for the tokens that need it, saving work on others. This is aligned with the broader trend in deep learning toward conditional and adaptive computation to handle varying input complexities efficiently. In conclusion, the background and literature on token pruning reveal a progression from simple static rules to sophisticated learned and hybrid strategies. Each class of methods addresses certain challenges. Heuristics are simple and fast but limited, RL-based methods were more flexible but costly, and differentiable methods offer a direct integration into training, which now seems the most promising path. Prompt-level methods extend these ideas to the new domain of large context LLMs. The background presented here lays the foundation for understanding the innovative approaches explored in this thesis. The following chapters will build on these concepts, diving into our proposed methodologies and how they improve upon the state of the art in token pruning for LLMs.

2.5

Read Between the Tokens: Differentiable Text Pruning via Perturbed Top- k Selection

Originally published as: Paolo Italiani, Luca Ragazzi, and Gianluca Moro. *Read between the tokens: Differentiable text pruning via perturbed top- k selection.* *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2026b

Abstract

Transformer-based pretrained language models (PLMs) face scalability issues due to their computational expense, which increases with the length of the input sequence, and often struggle to maintain focus on relevant information. To mitigate this, we introduce PRUNEPERT, a novel model featuring a learnable mechanism that identifies and removes uninformative tokens from the context. By doing so, our method not only addresses performance concerns but also enhances interpretability, offering valuable insights into the tokens utilized in the model’s decision-making process. Specifically, our approach employs a differentiable perturbed top- k token selection module within the transformer layers to prune a user-defined percentage of tokens. It can be integrated with any downstream PLMs, allowing them to be trained end-to-end using backpropagation. We demonstrate the application of PRUNEPERT in text summarization and classification tasks, utilizing both encoder-decoder PLMs and contemporary decoder-only large language models. Notably, our findings reveal that models equipped with PRUNEPERT achieve up to 2x higher throughput and exhibit comparable performance in text summarization, while demonstrating superior performance in text classification tasks.

2.5.1 Introduction

Pretrained language models (PLMs) have demonstrated remarkable efficacy across diverse natural language processing (NLP) tasks, including representational activities like text classification [Sun et al. \[2023\]](#) to generative ones such as summarization [Li et al. \[2024a\]](#). Despite these significant performance gains, PLMs often require substantial computational resources, posing challenges for deployment in real-world settings with low-memory GPUs [Moro and Ragazzi \[2022a, 2023\]](#). The high computational demand of PLMs is primarily due to the self-attention mechanism, which scales quadratically with sequence length. This becomes particularly burdensome when processing long texts [Beltagy et al. \[2020a\]](#), [Moro et al. \[2023f\]](#).

It is important to note that not all elements of the input text are equally significant [Goyal et al. \[2020\]](#). As tokens pass through the model layers, some are effectively encoded early on, rendering further processing redundant [Rogers et al. \[2021\]](#). As a result, removing these redundant tokens through token pruning can improve the model’s efficiency. Additionally, focusing model training on the most relevant text segments can enhance overall performance.

Over the years, various approaches have been introduced to advance research on token pruning in NLP. However, these solutions typically suffer from at least one of the following drawbacks: (i) They rely on reinforcement learning, which necessitates defining ad-hoc loss functions and undergoing an additional training phase [Ye et al. \[2021a\]](#). (ii) The module responsible for selecting crucial tokens depends on heuristics derived from attention scores and remains static, lacking the ability to learn [Goyal et al. \[2020\]](#), [Luo et al. \[2022a\]](#). (iii) They focus solely on accelerating the inference phase, ignoring the training stage [Yang et al. \[2023\]](#). (iv) They prioritize efficiency at the expense of model

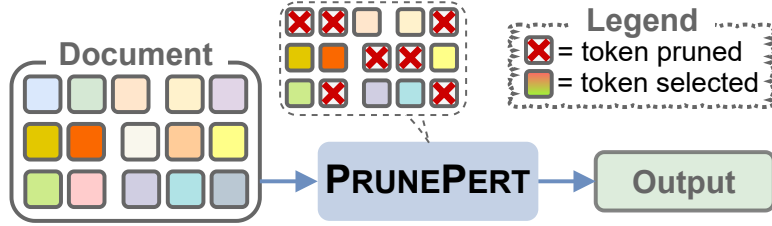


FIGURE 2.1. Overview of our PRUNEPERT model. In this work, “Output” can refer to either a summary or a class.

accuracy [Anagnostidis et al. \[2023\]](#).

To this end, we propose PRUNEPERT (Figure 2.1), a methodology that extends PLM architectures with a token-pruning layer. Specifically, we incorporate a differentiable perturbed top- k token selection mechanism within the transformer layers, aiming to identify a user-defined percentage of important input tokens in an end-to-end learning process. This method allows the model to process fewer input tokens, thereby enhancing the efficiency, interpretability, and accuracy of PLMs [Moro et al. \[2024b\]](#). We test our solution on two different NLP tasks: text summarization and classification. Through rigorous evaluations and ablation studies, we demonstrate that our solution generates coherent and informative summaries comparable to state-of-the-art baseline. Moreover, it achieves a 2x speed-up in computational time. When applied to the classification task, it yields up to a 2 percentage point improvement in the F1 score.

A conference version of this thesis was accepted at the Findings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024) [Ragazzi et al. \[2024b\]](#). In this manuscript, we extend our previous work by integrating a text classification task using four diverse datasets. We also implement PRUNEPERT on several large language models (LLMs) and introduce ablation studies. Furthermore, we conduct in-depth comparative experiments and expand the validation of our method. The main enhancements are detailed as follows:

- We assess the results on multiple classification datasets: BOOLQ [Clark et al. \[2019\]](#) for reading comprehension, BEAVERTAILS [Ji et al. \[2023\]](#) for safety alignment, ANLI [Nie et al. \[2020\]](#) for language understanding, and IMDB [Lan et al. \[2016\]](#) for sentiment analysis.
- We implement PRUNEPERT on three recent LLMs: MISTRAL-7B [Jiang et al. \[2023a\]](#), LLAMA-3-8B, GEMMA-7B [Team and Mesnard \[2024\]](#).
- We compare PRUNEPERT against two state-of-the-art token pruning techniques.
- We propose four architectural variants, each designed to improve the token scoring and selection mechanisms.
- We conduct ablation studies on the impact of hyperparameters related to the differentiable top- k token selection module, showing their effect on overall performance.

- Finally, we perform additional experiments and analyses on text summarization tasks.

2.5.2 Related Work

We begin by reviewing related research on token pruning techniques. Next, we discuss the downstream tasks selected for this study, including text summarization and classification.

2.5.2.1 Token Pruning

Preliminary strategies to reduce the computational demands of transformers, including those with linear complexity [Beltagy et al. \[2020a\]](#), [Huang et al. \[2021\]](#) and extract-then-abstract pipelines [Moro et al. \[2023b\]](#), [Ragazzi et al. \[2025\]](#), primarily focus on eliminating non-essential tokens by leveraging attention scores. PoWER-BERT [Goyal et al. \[2020\]](#) and LTP [Kim et al. \[2022\]](#) use a scoring mechanism derived from attention scores to selectively discard tokens based on their influence on subsequent computations. Extending this strategy to Vision Transformers (ViTs), [Luo et al. \[2022a\]](#) identify and prune tokens by analyzing attention scores and combining insights from multiple attention heads. Furthermore, [Yang et al. \[2022\]](#) assigns quantization precision levels to tokens (i.e., 0 bit, 4 bit, and 8 bit) based on their assessed significance, which is measured by their attention probabilities. Despite the promising results and widespread adoption, these techniques are fundamentally based on predefined heuristics. Consequently, we propose exploring an alternative approach that involves autonomously learning the token selection module. In this vein, Pyramid-BERT [Huang et al. \[2022b\]](#) introduces a theoretically grounded δ -cover core-set selection, solved via a parallelizable k-Center greedy algorithm with exponentially decaying per-layer lengths, enabling fixed-size end-to-end fine-tuning without auxiliary losses. AdapLeR [Modarressi et al. \[2022a\]](#) trains lightweight per-layer Contribution Predictors and employs a novel soft-removal masking function to gradually prune low-contributing tokens throughout the network. TR-BERT [Ye et al. \[2021a\]](#) treats token reduction as a multi-step challenge that can be solved through reinforcement learning, incorporating an additional loss function to optimize rewards. In contrast, [Cordonnier et al. \[2021\]](#) implement an end-to-end token selection strategy that avoids additional losses by using perturbation. This method, although differentiable, lacks sparsity. Innovatively, [Sander et al. \[2023\]](#) introduce the first differentiable and sparse top- k operator using p-norm regularization. However, these pruning-based approaches are limited to encoder-only architectures, restricting their applicability to generative tasks. In contrast, LLMLINGUA [Jiang et al. \[2023b\]](#) is designed for generative LLMs, introducing a coarse-to-fine prompt compression framework that combines a perplexity-based budget controller, an iterative token-level pruning mechanism, and distribution alignment to retain key content under extreme compression ratios, while also addressing distribution mismatch between the target LLM and the small language model used for prompt compression. LLMLINGUA-2 [Pan et al. \[2024b\]](#) improves upon its predecessor by introducing a length-bounded adaptive compression framework guided by a learnable reward model, enabling high-fidelity token selection

under strict length constraints. ICAE Ge et al. [2024] introduces an iterative compression and expansion framework where a small selector model prunes non-essential tokens and a corrector module reconstructs the original input, enabling efficient inference while preserving task-critical information.

2.5.2.2 Scientific Document Summarization (SDS)

To address the challenge of processing long inputs posed by lengthy scientific articles, prior work has relied on various solutions. These include linear attention mechanisms Beltagy et al. [2020a], Phang et al. [2023], segmentation-based techniques Gidiotis and Tsoumakas [2020], and marginalization-enhanced decoding Moro et al. [2022a, 2023e]. SDS has traditionally concentrated on creating technical summaries, such as abstracts. For example, Cohan et al. [2018] developed a method that uses a hierarchical encoder to capture the discourse structure of texts, along with an attentive discourse-aware decoder to facilitate summary generation. Similarly, An et al. [2021b] enhanced summarization techniques by integrating information from the source document and its referenced materials using a graph-based citation model.

In recent years, there has been a shift within the scientific community towards broadening the scope of text summarization to include text simplification. This paradigm shift has prompted the development of new datasets and methods aimed at bridging this gap between complex and simplified text summarization. A notable initiative in this domain is the LaySumm task Chandrasekaran [2020], which utilizes a corpus of 572 articles featuring author-generated lay summaries from diverse disciplines such as Materials Science, Archaeology, Hepatology, and Artificial Intelligence. Zaman et al. [2020] have contributed to this field by automatically retrieving simplified and condensed versions of articles from the Eureka Alert website. They amassed a dataset of 5204 instances from journals including PLOS-ONE, Nature Communication, and Scientific Reports. Additionally, Cachola et al. [2020] have proposed a novel approach to extreme summarization that focuses on distilling the core elements of texts while omitting detailed methodological descriptions. Furthermore, Guo et al. [2021a] compiled a significant collection of 6695 pairs of systematic reviews and their corresponding plain-language summaries from the Cochrane Database of Systematic Reviews, further enriching the resources available for research in this area.

2.5.2.3 Text Classification

Text classification is a task aimed at categorizing text into predefined categories, like sentiment polarity or topic. Earlier probabilistic approaches explored Markov chain-based methods for in-domain and cross-domain sentiment classification Domeniconi et al. [2015a]. Later, neural models employing character-based convolutional networks (CNNs) were introduced to capture local dependencies Zhang et al. [2015], Conneau et al. [2017]. To better capture long-term dependencies and higher-level textual features, long short-term memory networks were adopted due to their ability to maintain and update cell states over longer sequences. The integration of attention mechanisms by Wang et al.

[2016] facilitated the effective extraction of pivotal information from the entire text, regardless of its spatial position, by dynamically focusing on relevant parts of the input sequence. Building on these advancements, Yang et al. [2016] introduced a hierarchical architecture that allows for sentence-level attention focus, improving the model’s ability to capture hierarchical structures in text. In parallel, ensemble-based strategies have been explored to improve model robustness and cross-domain generalization, even in distinct fields such as bioinformatics Moro and Masseroli [2021]. To address the complexity and parameter count of these models, Joulin et al. [2017] proposed a streamlined approach using a basic linear model with a hidden layer, leveraging n-gram features and hierarchical softmax to enhance computational efficiency.

The introduction of PLMs, which consist of hundreds of millions of parameters and are pretrained on extensive corpora containing billions of tokens, has significantly advanced the field of NLP by enabling more accurate and efficient text representations Adhikari et al. [2019], Reimers and Gurevych [2019a]. More recently, the forefront of supervised text classification has been dominated by LLMs, which leverage vast amounts of data and complex architectures to achieve superior performance. Research indicates that LLMs consistently outperform traditional machine learning techniques across various classification tasks. Their robust generalizability allows them to adapt to new tasks without further training, making them highly versatile Brown et al. [2020]. To further optimize these models, they can be fine-tuned on the entire training corpus. This method enhances their performance and aligns more closely with traditional small language model frameworks, providing better task-specific adaptations Chae and Davidson [2023]. Several recent studies highlight the practical applications of LLMs: Törnberg [2023] utilized LLMs for classifying political messages on Twitter; Gilardi et al. [2023b] applied them to relevance and topic detection; and Huang et al. [2023a] leveraged LLMs for hate speech classification.

2.5.3 Method

We first provide the necessary theories and key methodologies essential for understanding our PRUNEPERT solution.

2.5.3.1 Preliminary

PRUNEPERT can be seamlessly integrated into any transformer-based architecture Vaswani et al. [2017b], including encoder-only, decoder-only, or encoder–decoder models. To ensure a comprehensive understanding, we describe our method within the context of encoder–decoder solutions. The model consists of an encoder $E = \{e_1, \dots, e_L\}$ and a decoder $D = \{d_1, \dots, d_L\}$, each comprising a stack of L identical layers. The encoder $E(\cdot)$ transforms the input sequence of symbols $x = (x_1, \dots, x_{|x|})$ into a series of continuous representations $\mathbf{h}_L = \{\mathbf{h}_{L,1}, \dots, \mathbf{h}_{L,|x|}\} \in \mathbb{R}^{|x| \times d}$, effectively capturing the input’s contextual information, where d is the number of hidden dimensions. Using these representations, the decoder $D(\cdot)$ generates the output sequence $\mathbf{y} = \{\mathbf{y}_1, \dots, \mathbf{y}_{|y|}\}$ in an autoregressive manner, where each token is generated sequentially based on the previously

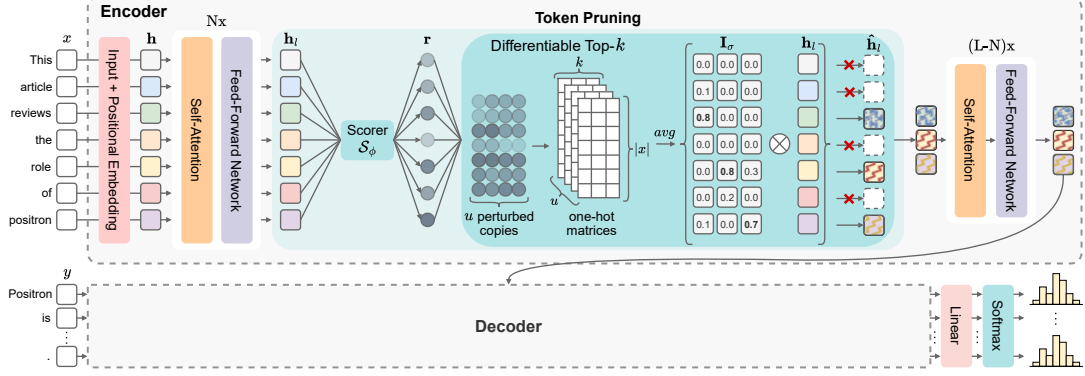


FIGURE 2.2. PRUNEPERT architecture. In this example (for encoder–decoder models), the number of selected tokens is $k = 3$ and the number of perturbed copies is $u = 4$.

generated tokens. Each encoder layer e_i includes a self-attention module (SELF-ATT) and a feedforward network module (FFN), both connected through residual connections and layer normalization (LN):

$$\begin{aligned} \mathbf{m}_{i-1} &= \text{LN}(\mathbf{h}_{i-1} + \text{SELF-ATT}(\mathbf{h}_{i-1})), \\ \mathbf{h}_i &= \text{LN}(\mathbf{m}_{i-1} + \text{FFN}(\mathbf{m}_{i-1})). \end{aligned} \quad (2.1)$$

Additionally, the decoder $D(\cdot)$ incorporates a cross-attention module (CROSS-ATT) between the SELF-ATT and FFN modules, performing multi-head attention over the encoder’s output $\mathbf{h}_L$, which enables the decoder to effectively attend to and integrate contextual information from the entire input sequence.

$$\mathbf{c}_{i-1} = \text{LN}(\mathbf{m}'_{i-1} + \text{CROSS-ATT}(\mathbf{m}'_{i-1}, \mathbf{h}_L)), \quad (2.2)$$

where $\mathbf{m}'_{i-1}$ is the output of decoder’s SELF-ATT after being processed by LN and summed with the residual connection.

2.5.3.2 Perturbed Token Pruning

We present PRUNEPERT (see Figure 2.2), a model that employs token-level pruning in the encoder through a differentiable perturbed top- k module. Our approach facilitates end-to-end training without the need for auxiliary losses.

Model Architecture Our model uses a transformer to leverage the knowledge embedded in PLMs. Within the encoder E , we incorporate a scorer network $\mathcal{S}_\phi$ and a token selection module $\mathcal{T}$, where $\phi \in \mathbb{R}^d$ denotes the trainable parameters of $\mathcal{S}_\phi$, which are the only additional parameters introduced by our method. These components are positioned between layers e_l and e_{l+1} of E , with l as a hyperparameter.

The continuous token representations $\mathbf{h}_l$ are input into $\mathcal{S}_\phi$, which calculates relevance scores $\mathbf{r} = \mathcal{S}_\phi(\mathbf{h}_l) \in \mathbb{R}^{|x|}$ for each token in the input sequence x . Then, the token

selection module $\mathcal{T}$, a discrete operator, identifies the indices $\mathbf{i} = \mathcal{T}(\mathbf{r}) \in [1, |x|]^k$ of the k most salient tokens.¹ These indices $\mathbf{i}$ must be sorted in ascending order (i.e., $\mathbf{i}_1 < \mathbf{i}_2 < \mathbf{i}_{z+1} < \mathbf{i}_k$) to maintain the original sequence order of the hidden states.

Conceptually, if one imagines a large tensor containing all token embeddings, the relevant embeddings could be efficiently extracted through a single matrix multiplication operation. To facilitate this extraction, each index $\mathbf{i}_z$ is represented as a $|x|$ -dimensional one-hot vector $\mathbf{I}_z$. These vectors compile into the matrix $\mathbf{I} = \mathbf{I}_1, \dots, \mathbf{I}_k \in 0, 1^{|x| \times k}$. Consequently, the pruned hidden states $\hat{\mathbf{h}}_l$ are derived by multiplying the transpose of $\mathbf{I}$, denoted as $\mathbf{I}^\top$, with $\mathbf{h}_l$.

Differentiable Top- k The token selection module $\mathcal{T}$ is inherently non-differentiable, preventing the backpropagation of gradients necessary to adjust the parameters ϕ of the scorer network $\mathcal{S}$. To enable the differentiation of this module and facilitate the learning of ϕ , we adopt the perturbed maximum method as outlined in Berthet et al. [2020], which allows gradients to flow through the non-differentiable selection process. The forward pass of this method is defined as follows:

$$\mathbf{I}_\sigma = \mathbb{E} \left[\arg \max_{\mathbf{I} \in \mathcal{C}} \langle \mathbf{I}, \mathbf{r} \mathbf{1}^\top + \sigma \mathbf{Z} \rangle \right], \quad (2.3)$$

where $\mathbf{r} \mathbf{1}^\top \in \mathbb{R}^{|x| \times k}$ represents the scores replicated k times, σ is a hyperparameter that controls the impact of the noise introduced during the selection process, and $\mathcal{C}$ represents the constraint set, defined as follows:

$$\begin{aligned} \mathcal{C} = \{ & \mathbf{I} \in \mathbb{R}^{|x| \times k}, \mathbf{I}_{a,b} \geq 0, \mathbf{1}^\top \mathbf{I} = \mathbf{1}, \\ & \mathbf{I} \mathbf{1} \leq \mathbf{1}, \sum_{a \in [|x|]} a \mathbf{I}_{a,k^*} < \sum_{b \in [|x|]} b \mathbf{I}_{b,k'} \forall k^* < k' \}. \end{aligned} \quad (2.4)$$

The condition $\mathbf{1}^\top \mathbf{I} = \mathbf{1}$ ensures that each column in the matrix $\mathbf{I}$ sums to 1, guaranteeing that exactly one token is selected per column. The requirement for ordered $\mathbf{i}$ indices is upheld by the last constraint, ensuring the sequential integrity of the token selection process.

In practice, we introduce perturbations to $\mathbf{r} \mathbf{1}^\top$ by sampling u instances of Gaussian noise $\mathbf{Z}$. For each instance of perturbed input, we apply the top- k selection algorithm. We then estimate the expected outcome of Equation 2.3 using a Monte-Carlo method, averaging the results from each perturbed scenario. As detailed by Berthet et al. [2020], the Jacobian matrix associated with the described forward pass is determined as follows:

$$J = \mathbb{E} \left[\arg \max_{\mathbf{I} \in \mathcal{C}} \langle \mathbf{I}, \mathbf{r} \mathbf{1}^\top + \sigma \mathbf{Z} \rangle \mathbf{Z}^\top / \sigma \right]. \quad (2.5)$$

Applying the top- k operator to each perturbed input generates the one-hot matrix $\mathbf{I}$. However, the average of these matrices often deviates from the one-hot configuration,

¹ $\mathbf{i}_1 = z$ implies that the first selected token is the z -th of the input x .

TABLE 2.1. Data statistics for the scientific document summarization task. The number of words and sentences are averaged.

Dataset	Samples	Doc	Tech Summary		Lay Summary	
		# words	# words	# sents	# words	# sents
SciLAY	43,790	7530.4	239.1	8.8	145.7	5.7

TABLE 2.2. The statistics of the dataset used for text classification. All reported values are averaged across all instances except for the number of samples and classes.

Dataset	Samples	Source		Target
		# words	# sents	classes
BOOLQ Clark et al. [2019]	12,697	117.3	5.0	[True 62.3%, False 37.7%]
BEAVERTAILS Ji et al. [2023]	30,206	83.4	4.4	[Safe 44.6%, Not Safe 55.4%]
ANLI Nie et al. [2020]	18,946	80.1	3.8	[Entail 31.7%; Contradict 26.7%; Neutral 41.6%]
IMDB Lan et al. [2016]	25,000	282.6	10.8	[Positive 50.0%, Negative 50.0%]

especially in the early stages of training when the scoring system is not yet stabilized, resulting in indecision. Consequently, the pruned k hidden states $\hat{\mathbf{h}}_l$ are computed as a weighted average of the original hidden states $\mathbf{h}_l$. This process yields a significant advantage: the backpropagated gradients are influenced by all tokens, not just those initially selected. This eliminates the need for multiple training iterations to consistently sample the most appropriate tokens. During inference, we enhance efficiency by employing a hard top- k selection, eliminating the need for u perturbed repetitions. This approach, however, introduces a discrepancy between training and testing conditions. To mitigate this train-test gap, we follow the strategy recommended by Cordonnier et al. [2021], linearly decreasing σ to 0 during training. This adjustment ensures that by the end of training, no noise is added, and the behavior of the differentiable top- k operator aligns numerically with the hard top- k operator.

Note: We implement PRUNEPERT within the encoder layers of encoder-decoder models for the text summarization task. For text classification tasks using LLMs, PRUNEPERT is integrated into the decoder layers. This strategic placement ensures that our pruning method aligns optimally with the specific requirements and architectures of various NLP tasks.

2.5.4 Experimental Setup

2.5.4.1 Datasets

Statistical details of the datasets are given in Table 2.1 for the summarization task and in Table 2.2 for the classification task.

TABLE 2.3. Split statistics of the journals within SciLay.

Split	Website	Samples	Doc	Tech Summary		Lay Summary	
			# words	# words	# sents	# words	# sents
NC	https://www.nature.com/ncomms/	6937	8906.4	168.3	6.5	47.7	2.1
PLGEN	https://journals.plos.org/plosgenetics/	3859	9554.3	257.6	9.5	195.0	7.7
PLPAT	https://journals.plos.org/plospathogens/	3650	9569.2	260.7	9.7	196.3	7.7
PLCB	https://journals.plos.org/ploscompbiol/	3237	9683.9	254.7	9.3	192.8	7.5
PLNTD	https://journals.plos.org/plosntds/	2862	6341.0	304.5	10.2	198.9	7.9
PLB	https://journals.plos.org/plosbiology/	1121	10165.1	247.3	9.1	216.7	8.0
B	https://www.mdpi.com/journal/biology	2022	5854.4	246.8	9.2	155.9	6.1
CB	https://www.nature.com/commsbio/	1084	8462.0	170.0	6.9	55.4	2.9
SD	https://www.nature.com/sdata/	907	5355.0	180.6	6.8	45.3	1.0
MBIO	https://journals.asm.org/journal/mbio	759	8403.5	246.5	9.1	145.0	5.7
A	https://www.mdpi.com/journal/animals	4887	5494.3	263.1	9.4	163.6	6.3
I	https://www.mdpi.com/journal/insects	1477	5372.2	238.0	9.1	169.7	6.9
C	https://www.mdpi.com/journal/cancers	8478	5851.8	242.4	9.3	133.4	5.3
OTHER	-	2510	7427.9	262.2	9.1	173.6	6.0

Text Summarization The summarization capabilities of our model are assessed using the **SciLay** dataset,² a comprehensive collection of documents from various natural science journals. Each document includes both technical summaries, rich in specialized terminology, and lay summaries, written in accessible language with contextual explanations for better understanding. Table 2.3 provides the dataset split statistics.

Text Classification The classification efficacy of our model is evaluated using a variety of datasets. **BoolQ** Clark et al. [2019] is a reading comprehension dataset that includes naturally posed yes/no questions. **BeaverTails** Ji et al. [2023] aims to enhance research on safety alignment in LLMs and includes human-labeled question-answer (QA) pairs categorized into various harm categories. **ANLI** Nie et al. [2020] presents a new challenge for natural language understanding, with each instance containing a context, a hypothesis, and a target label of either “entailment”, “contradiction”, or “neutral”. **IMDB** Lan et al. [2016] consists of movie reviews from the Internet Movie Database, including only highly polarized opinions: negative reviews (ratings of 4 or below) and positive reviews (ratings of 7 or above) on a scale of 10. Neutral reviews are omitted from this dataset.

2.5.4.2 Baselines

Text Summarization We use two encoder-decoder models: **Bart** Lewis et al. [2020c] and **Pegasus** Zhang et al. [2020a]. BART is renowned for its denoising pretraining objective and has become a popular choice in various NLP applications. PEGASUS is distinguished by its pretraining objective specifically designed to enhance abstractive summarization capabilities. To accommodate input sequences exceeding 1024 tokens, we use LSG Condevaux and Harispe [2023] for both models. This architecture enables the processing of extended sequences with linear computational complexity ($\mathcal{O}(n)$) relative

²<https://huggingface.co/datasets/disi-unibo-nlp/SciLay>

to the input length, where n is the number of tokens.

Text Classification We employ several LLMs, each demonstrating unique advancements in different areas. **Mistral-7B** Jiang et al. [2023a] excels across all evaluated benchmarks compared to LLAMA-2-13B Touvron et al. [2023c], and outperforms LLAMA-34B Touvron et al. [2023a] in tasks involving mathematical reasoning and code generation. **LLaMA-3-8B** significantly improves upon its predecessor, LLAMA-2, setting a new standard for LLMs of this scale through enhanced architecture and training techniques. **Gemma-7B** Team and Mesnard [2024], part of the GEMINI model family Team et al. [2023], was trained on an extensive dataset of 6 trillion tokens of text.

For each task, we compare baseline models with their counterparts augmented by PRUNEPERT (abbreviated as PP).

2.5.4.3 Evaluation Metrics

Text Summarization To thoroughly evaluate model performance, we consider multiple dimensions. First, we present the F1 scores for syntactic metrics, specifically **ROUGE-1,2,L** Lin [2004] (R-1,2,L), and introduce a variance-aware aggregated score, $\mathcal{R} = \text{avg}(r_1, r_2, r_L) / (1 + \sigma_r^2)$, where σ_r^2 is the F1 variance Moro et al. [2023a]. Second, we use **BARTScore** Yuan et al. [2021a] (BaS), a model-based metric, to evaluate semantic coverage. We report precision, recall, and F1 scores to measure the model’s ability to capture and reproduce the semantic content of reference texts. We also evaluate readability of the lay summaries using Flesch-Kincaid Grade Level (**FKGL**, Kincaid et al. [1975b]), Coleman-Liau Index (**CLI**, Coleman and Liau [1975a]), Dale-Chall Readability Score (**DCRS**, Dale and Chall [1948]), and Gunning Fog Index (**GF**, Gunning [1952]). These metrics estimate the education level required to comprehend a text, with lower scores indicating higher readability. Scores between 13 and 16 correspond to college-level reading proficiency in the US. Finally, we conduct an extensive **human evaluation** to gain insights into the subjective quality and relevance of the model outputs from the perspective of users. We also measure model efficiency by monitoring *training runtime*, which helps assess the practicality and scalability of each model in real scenarios.

Text Classification To assess model performance on the classification task, we use macro-averaged and micro-averaged F1 scores Blair [1979], referred to as **F1-M** and **F1-m**. F1-M is the arithmetic mean of the F1 scores for each class, treating all classes equally regardless of their frequency in the dataset. Conversely, F1-m aggregates the counts of true positives, false positives, and false negatives across all classes before calculating the F1 score. This method provides a global average, reflecting the model’s overall accuracy and giving more weight to more frequently represented classes.

2.5.4.4 Hardware Setup and Environment

All experiments are monitored using Wandb and conducted on a workstation equipped with a single Nvidia GeForce RTX3090 GPU, which has 24 GB of VRAM and 64 GB

TABLE 2.4. Hyperparameters used. $\dagger$ indicates values specific to PRUNEPERT. * denotes the final chosen value. $0.5 \times |x|$ indicates selecting half of the input tokens.

Hyperparameter	Pegasus	LLMs
Dropout rate	0.1	0.1
Learning rate	5e-5, linear scheduler	5e-5, linear scheduler
Optimizer	$0.9 \beta_1, 0.999 \beta_2, 1e-2$ weight decay	$0.9 \beta_1, 0.999 \beta_2, 1e-2$ weight decay
Batch size	1	1
Epochs	1	1
Decoding strategy	greedy search	greedy search
Seed	42	42
$k^\dagger$ selected tokens †	$\{0.1, \dots, 0.5^*, \dots, 0.9\} \times x $	$\{0.3, 0.4, 0.5, 0.6, 0.7, 0.8^*, 0.9\} \times x $
l encoder layer †	$\{1, 2, 3^*, \dots, 15\}$	$\{10, 15, 20^*, 25, 30\}$
u sampled noises †	$\{50, 100, 200^*, 300, 400\}$	$\{20, 40, 60, 80, 100^*, 120, 140\}$
$\sigma^\dagger$	$\{0.05, 0.1, 0.2^*, 0.3\}$	$\{0.05, 0.1^*, 0.15, 0.2, 0.25\}$

TABLE 2.5. Overall results on SciLay. Best scores are bolded. * means statistical significant improvement w.r.t. baseline PEGASUS ($p < 0.05$ with student t-test). Violet: Our models.

Model	R-1	R-2	R-L	$\mathcal{R}$	BaS-R	BaS-P	BaS-F1	Read.
Technical Summarization								
BART	28.70	4.95	16.07	16.42	-3.363	-4.887	-4.125	-
PEGASUS	35.01	10.06	25.40	23.24	-3.380	-2.333	-2.856	-
PEGASUS-LLMLINGUA-2	31.25	6.81	19.57	19.02	-3.413	-4.053	-3.733	-
PEGASUS-PRUNEPERT	37.61*	9.35	26.27*	24.09*	-3.663	-2.403	-3.033	-
Lay Summarization								
BART	23.86	3.99	15.57	14.38	-3.299	-5.094	-4.196	16.67
PEGASUS	40.76	12.69	28.57	26.98	-3.258	-3.228	-3.243	16.08
PEGASUS-LLMLINGUA-2	40.36	12.51	28.82	26.88	-3.278	-2.904	-3.091	16.27
PEGASUS-PRUNEPERT	38.32	10.93	27.23	25.17	-3.451	-2.776*	-3.113*	15.29

of system RAM. The system also includes an Intel® Core™ i9-10900X CPU operating at 3.70GHz. Ubuntu 20.04.3 LTS serves as the operating system. For each baseline, we conduct experiments on both the standard version and its token pruning variants PRUNEPERT, LLMLINGUA, and LLMLINGUA-2. In the summarization experiments, however, we exclude LLMLINGUA due to its high preprocessing cost (approximately 8 seconds per instance), which makes it impractical to apply across the full dataset. We list the hyperparameters used for fine-tuning and inference in Table 4.13. For fairness, the target number of selected tokens k in LLMLINGUA and LLMLINGUA-2 is matched to that of PRUNEPERT.

TABLE 2.6. Results of our Pegasus-PP model on the text summarization task across the journals in SciLay. Improved performance by PRUNEPERT are shown in bold. Values in parentheses indicate the performance gap relative to the baseline Pegasus model. * denotes statistically significant gap.

Lay Summarization				Technical Summarization			
R-1	R-2	R-L	BaS-F1	R-1	R-2	R-L	BaS-F1
Nature Communications							
37.10 (-1.73*)	13.02 (-1.53*)	31.88 (-0.84*)	-3.29 (+0.04)	37.47 (+0.72*)	8.67 (-2.27*)	26.11 (-0.10)	-3.171 (-0.12*)
Plos Genetics							
39.26 (-5.99*)	10.39 (-2.66*)	27.06 (-1.19*)	-3.11 (+0.15*)	37.56 (+1.01*)	9.32 (-0.39)	25.96 (+0.18)	-3.11 (-0.18*)
Plos Pathogens							
40.50 (-4.78*)	10.88 (-2.25*)	27.83 (-0.47)	-3.01 (+0.16*)	37.02 (+1.33*)	9.18 (-0.44)	25.55 (+0.16)	-3.04 (-0.20*)
Plos Computational Biology							
38.19 (-5.34*)	9.89 (-1.59)	26.76 (-0.84*)	-3.28 (+0.17*)	39.80 (+3.22*)	9.97 (+0.58*)	27.12 (+1.59*)	-3.17 (-0.07*)
Plos Neglected Tropical Diseases							
41.43 (-2.41*)	11.28 (-1.50*)	28.42 (+0.49)	-2.93 (+0.20*)	37.26 (+2.45*)	8.72 (-1.08*)	25.36 (+0.88*)	-2.97 (-0.20*)
Plos Biology							
36.88 (-8.23*)	10.15 (-1.92*)	26.24 (-1.13)	-3.16 (+0.18*)	38.17 (+3.84*)	9.99 (+1.01*)	26.50 (+1.93*)	-3.11 (-0.20*)
Biology							
37.55 (-1.85*)	10.68 (-1.15)	27.28 (-0.13)	-3.08 (+0.16*)	36.34 (+3.86*)	9.07 (+0.13)	26.26 (+1.86*)	-3.00 (-0.20*)
Communications Biology							
31.47 (-1.37)	8.44 (-1.36)	25.16 (-0.87)	-3.31 (-0.05)	37.23 (-0.59)	8.25 (-3.57*)	25.92 (-1.48)	-3.08 (-0.14)
Scientific Data							
34.09 (-1.92)	20.24 (-3.07)	38.46 (-2.13)	-3.75 (-0.08)	36.44 (+2.07*)	7.74 (-1.47)	25.04 (+0.47)	-3.22 (-0.08)
mBio							
33.67 (-8.08*)	8.61 (-3.22*)	25.41 (-1.78*)	-3.21 (+0.06)	37.35 (-0.62)	9.11 (-1.80*)	25.37 (-0.83)	-3.08 (-0.21*)
Animals							
39.93 (-0.92)	11.89 (-0.30)	28.61 (+1.09*)	-3.06 (+0.19*)	38.92 (+6.57*)	10.31 (+0.48*)	27.66 (+2.56*)	-2.96 (-0.29*)
Insects							
38.51 (-1.33)	10.41 (-0.84)	27.58 (+0.97)	-3.17 (+0.12*)	37.01 (+5.41*)	8.55 (+0.01)	26.30 (+2.17*)	-3.11 (-0.28*)
Cancers							
38.42 (-1.44*)	11.82 (-0.69*)	28.48 (+0.52)	-2.88 (+0.18*)	37.27 (+3.05*)	9.95 (-0.49)	26.36 (+1.02*)	-2.84 (-0.15*)

TABLE 2.7. Overall results on the classification task. For each model, the best scores are in bold. Violet: Our models.

Model	BOOLQ		BEAVERTAILS		ANLI		IMDB	
	F1-M	F1-m	F1-M	F1-m	F1-M	F1-m	F1-M	F1-m
MISTRAL	87.49	88.07	85.20	85.56	71.17	71.40	96.40	96.40
MISTRAL-LLMLINGUA	84.37	85.50	84.84	85.17	66.57	66.80	95.03	95.04
MISTRAL-LLMLINGUA-2	85.23	85.75	85.45	85.63	67.98	68.10	96.24	96.24
MISTRAL-PRUNEPERT	89.39	90.03	86.43	86.69	72.74	72.90	96.68	96.68
LLAMA-3	87.60	88.38	83.88	84.24	68.68	68.80	96.84	96.84
LLAMA-3-LLMLINGUA	83.63	84.77	84.01	84.24	67.75	67.80	94.83	94.84
LLAMA-3-LLMLINGUA-2	87.68	88.32	85.45	85.70	64.19	64.20	96.64	96.64
LLAMA-3-PRUNEPERT	87.58	88.44	84.85	85.17	69.71	69.80	96.84	96.84
GEMMA	90.09	90.70	85.67	85.89	70.69	70.80	96.80	96.80
GEMMA-LLMLINGUA	85.13	86.36	86.10	86.36	70.03	70.30	96.68	96.68
GEMMA-LLMLINGUA-2	87.89	88.62	85.45	85.70	65.06	65.30	97.32	97.32
GEMMA-PRUNEPERT	89.90	90.52	85.84	86.16	70.82	70.90	96.84	96.84

2.5.5 Discussion

2.5.5.1 Effectiveness

Text Summarization Table 2.5 presents the performance outcomes on SCILAY. Overall, the PRUNEPERT variant performs comparably to the standard PEGASUS baseline. Additionally, PRUNEPERT improves the readability of lay summaries, as reflected by the average readability scores across the four metrics discussed in Section 2.5.4.3. We also observe that PRUNEPERT markedly outperforms LLMLINGUA-2 in the technical summarization task, whereas the reverse pattern emerges in the lay setting, albeit with considerably smaller differences. These results suggest that PRUNEPERT constitutes a more robust approach overall. Furthermore, we notice that PEGASUS-based solutions demonstrate superior skill in producing lay summaries. This can be attributed to the increased complexity in generating technical summaries, which demands greater length and lexical sophistication.

Table 2.6 presents a comprehensive analysis of the experimental summarization results, detailing individual scores for each of the 13 journals included in SCILAY. Consistent with the overall dataset results, our approach shows lower ROUGE scores for the lay summarization task but higher BartScore-F1 scores. Conversely, the technical summarization task shows an opposite trend. More precisely, for the lay summarization task, our methodology yields lower ROUGE-1 and ROUGE-2 scores across all journals. However, we observe higher ROUGE-L scores for 4 journals and improved BartScore for 10 journals. Conversely, for the technical summarization task, our solution underperforms in BartScore across all journals. However, it achieves higher scores in ROUGE-1, ROUGE-2, and ROUGE-L for 11, 5, and 10 journals, respectively.

Text Classification Table 2.7 presents the performance outcomes on the classification datasets. The application of PRUNEPERT substantially enhances the performance of the backbone models across nearly all datasets. The improvements are especially notable for MISTRAL-7B, with an average increase of 1.25 and 1.22 for F1-M and F1-m scores, respectively. In contrast, the performance gains for LLAMA-3-8B and GEMMA-7B, as well as for the IMDB dataset in general, are less pronounced. This is attributed to these backbone models approaching their performance ceiling, having already captured most discernible patterns, thus leaving limited scope for further significant enhancements. Additionally, PRUNEPERT consistently outperforms both LLMLINGUA and LLMLINGUA-2, with the sole exceptions of LLAMA-3 on BEAVERTAILS and GEMMA on IMDB when paired with LLMLINGUA-2. Yet, the improvements over PRUNEPERT are marginal. Overall, PRUNEPERT proves more robust across all settings, whereas the LLMLINGUA variants exhibit pronounced performance drops, most notably on ANLI.

2.5.5.2 Efficiency

Text Summarization In this study, our objective extends beyond assessing summarization efficacy to include reducing the computational burden during training. Fig-

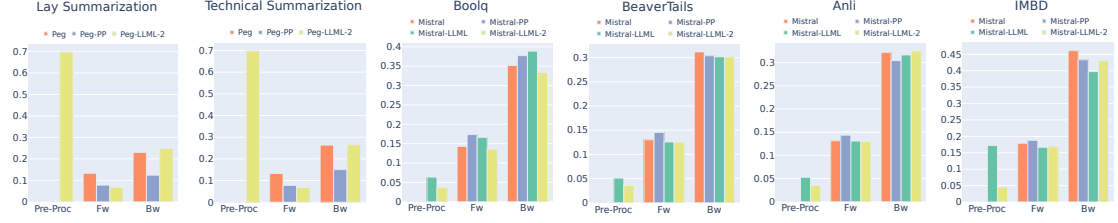


FIGURE 2.3. Average execution time (seconds) per instance during pre-processing (Pre-Proc), forward (fw), and backward (bw) passes for summarization on SciLay and classification with MISTRAL-7B. Peg-PP: Pegasus-PrunePert; LLML: LLMingua.

Figure 2.3 demonstrates the considerable speedup attained by integrating PRUNEPERT into PEGASUS. This efficiency improvement is observable in both the forward and backward propagation phases, resulting in gains ranging from 42% to 46%. The higher improvement is observed during the backward phase, which inherently demands more computational resources. By contrast, LLMLINGUA-2 yields only negligible improvements in forward-pass time, while the backward passes remain essentially unchanged relative to the baseline. Moreover, the method incurs substantial preprocessing overhead, which renders it considerably less practical than PRUNEPERT.

Text Classification Unlike summarization tasks, where models process lengthy texts (see Table 2.1), the documents in classification datasets contain significantly fewer words (see Table 2.2). When testing LLMs, the computational benefits during cross-attention computation are not realized in both the encoder and decoder, as in summarization. For these reasons, our primary goal is to enhance the performance of the backbone model. As indicated in Table 4.13, we selected values of k (80% of input tokens) and l (the token pruning module placed after the 20th decoder layer) to maximize the model’s accuracy, without prioritizing efficiency. Figure 2.3 demonstrates that the backward and forward times of MISTRAL-7B are not markedly different from those of MISTRAL-PRUNEPERT. Similarly, the LLMLINGUA variants, evaluated under the same selection ratio, exhibit comparable computational costs, though with the added overhead of a preprocessing step.

2.5.5.3 Interpretability

To gain deeper insights into the operation of the token pruning module, it is essential to examine both the tokens it preserves and eliminates. Token pruning enhances interpretability by functioning as an implicit rationalization process, where the model selectively retains tokens most relevant to its prediction. This mechanism aligns with recent rationalization frameworks that emphasize joint optimization of task performance and explanation fidelity [Ruggeri and Signorelli \[2025\]](#), [Liu et al. \[2022\]](#). The preserved subset effectively reveals the evidence the model relies on, offering a faithful and human-interpretable rationale. Because the pruning operates within the model’s computation rather than as a post-hoc step, it yields more faithful explanations of the decision pro-

cess. In our investigation, we focus on analyzing the occurrence frequencies of specific stopwords and parts-of-speech (POS) tags, including nouns, verbs, and punctuation, using ScispaCy [Neumann et al. \[2019\]](#). Furthermore, we examine the distribution of tokens relative to their positions in the text.

Text Summarization Figure 2.5 shows the results of our investigation into lay and technical summarization, illustrating the mean occurrence rates of each POS tag and stopword among both selected and unselected tokens within the source article. As anticipated, we note a propensity to retain significant informational content encoded in nouns and verbs. Conversely, elements essential for grammatical coherence but providing limited informational value are omitted. This observation holds true for both lay and technical summarization. However, technical summarization exhibits a higher tendency to retain stopwords, presumably to enhance syntactic overlap.

Figure 2.7 highlights a significant difference in token positioning between technical and lay summaries. In lay summaries, more tokens are concentrated in the initial 25% of the input, while in technical summaries, tokens are more concentrated in the later segments. However, token distribution in the last 75% of the input is relatively consistent across both tasks, with no notable differences. The introduction traditionally serves as the preamble in an article. Thus, the need for additional contextual information in a lay summary means the selection process is significantly influenced by the introductory section.

Text Classification Figure 2.5 shows intriguing patterns in the distribution of POS tags and stopwords in classification tasks using MISTRAL-PRUNEPERT. Our token pruning module behaves distinctly compared to in the summarization scenario. Notably, nouns are discarded more frequently, while punctuation impacts the prediction accuracy. This difference occurs because the goal here is to enhance model accuracy, not to speed up processing. Therefore, preserving the grammatical structure of key sentences and removing misleading text elements is crucial for accurate model predictions.

Figure 2.7 shows that the start-of-sequence (SOS) token is consistently important across all datasets. This is expected since the SOS token encapsulates the meaning of the entire input text. Notably, both the BEAVERTAILS and BOOLQ datasets assign significant importance to the last token, with BOOLQ emphasizing it more than preceding tokens. This is likely because, like the SOS token, the last token encapsulates important information, especially as the BOOLQ dataset often places the question in the final sentence. Similarly, for ANLI, the latter part of the input is significant because the hypothesis is placed at the end. However, here, greater importance is attributed to the text itself rather than the final token.

2.5.5.4 Human Evaluation

We conducted a thorough qualitative analysis of summaries produced by PEGASUS and PEGASUS-PRUNEPERT through a comprehensive human evaluation study. We randomly

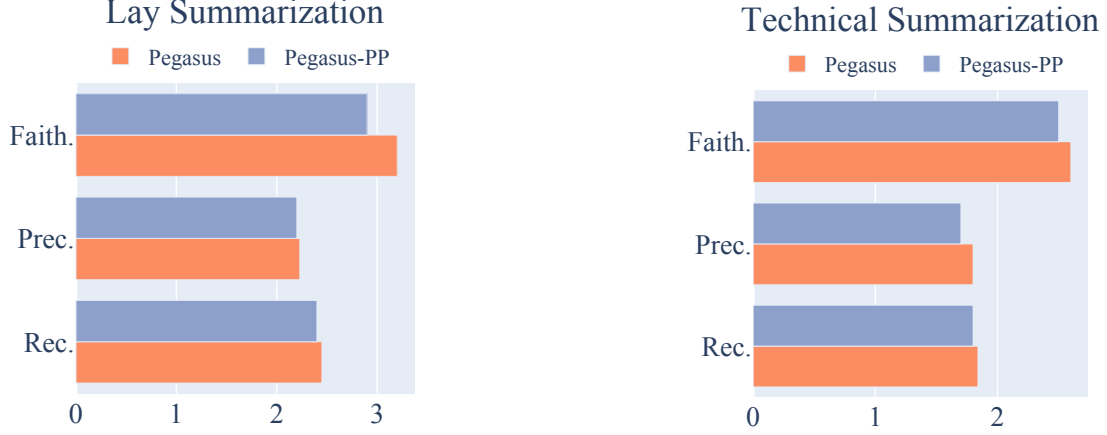


FIGURE 2.4. Human evaluations results considering faithfulness (Faith.), precision (Prec.), and recall (Rec.).

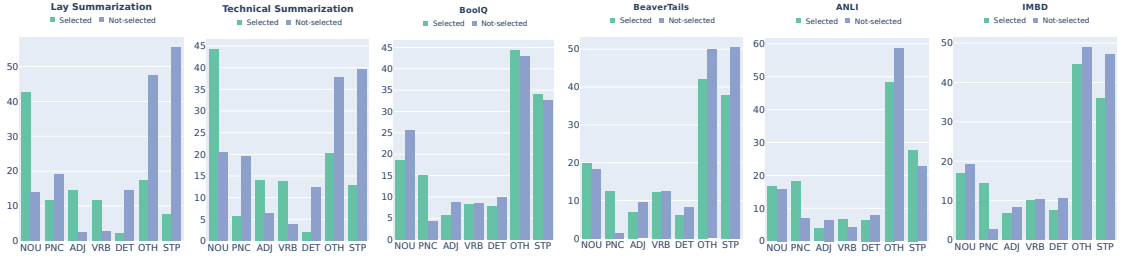


FIGURE 2.5. Average occurrence rates in order: nouns, punctuation, adjectives, verbs, determiners, other categories, and stopwords.

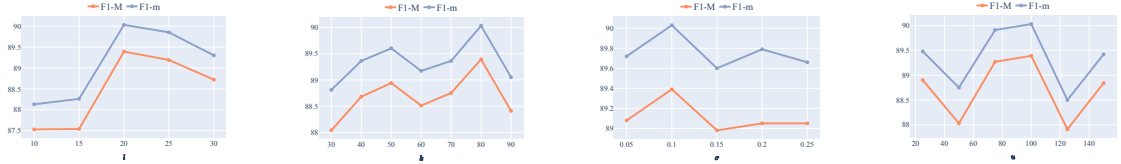


FIGURE 2.6. Differentiable top-k hyperparameters impact on model performance on BoolQ classification dataset with MISTRAL-7B.

selected 30 instances from the SciLAY test set and enlisted three proficient English annotators with strong NLP skills. Each annotator reviewed the articles and their corresponding target summaries. They then scored the generated summaries using a Likert scale from 1 (worst) to 5 (best) across three dimensions: (i) **recall**, evaluates if the generated summary encapsulates all essential contents of the target; (ii) **precision**, assesses if the generated summary includes only pertinent contents without extraneous information; (iii) **faithfulness**, checks if the generated summary maintains factual consistency with the original input document. The results of this study, depicted in Figure 4.4, show no discernible distinction among the solutions.

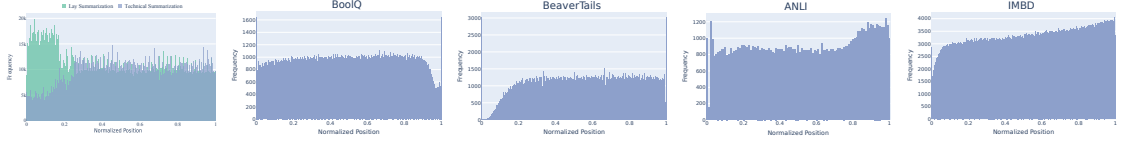


FIGURE 2.7. Normalized position distribution of the selected tokens from PRUNEPERT on the summarization and classification tasks.

2.5.5.5 Differentiable Top-k Hyperparameters Impact

The differentiable perturbed top- k module, detailed in Section 2.5.3.2, is defined by hyperparameters σ , u , l , and k . These hyperparameters significantly affect overall performance. We conducted an experiment on BOOLQ with MISTRAL-7B to examine how varying these hyperparameters affects the F1 score. Figure 2.6 shows that σ has the smallest impact on classification accuracy, with only a 0.4 percentage point difference between its best and worst values. In contrast, the other three hyperparameters show a substantial impact, with a 1.4 percentage point improvement from the worst to best values. As expected, the F1 score increases with higher l and k values. However, unexpectedly, the scores peak at 20 and 80%, respectively, then decline beyond these thresholds. The optimal number of sampled noises u is 75 and 100, with 100 yielding peak performance. Values outside this range, whether lower or higher, result in performance decline.

2.5.5.6 Architectural Variants

Based on the model architecture described in 2.5.3.2, we propose four variants applied to our best-performing backbone model, MISTRAL. In **Attn** we omit the learnable scorer network $\mathcal{S}_\phi$ and instead infer token significance from attention scores $A \in \mathbb{R}^{a_h \times |x| \times |x|}$ from the l -th layer, average them across all attention heads to get $\bar{A} \in \mathbb{R}^{|x| \times |x|}$, and then compute the average of the rows of $\bar{A}$ to obtain $\mathbf{r}$. In **Ens** we use an ensemble scoring mechanism with two scorer networks, $\mathcal{S}_{\phi_1}$ and $\mathcal{S}_{\phi_2}$. The scores $\mathbf{r}_1$ and $\mathbf{r}_2$ are averaged to produce the final relevance scores, $\mathbf{r}$. In **Prog** the text sequence is progressively compressed as the architecture deepens, retaining a balanced input representation and avoiding drastic performance drops. We use two differentiable token selection modules and two scorer networks, $\mathcal{S}_{\phi_1}$ and $\mathcal{S}_{\phi_2}$, positioned between layers e_{l_1} , e_{l_1+1} , and e_{l_2} , e_{l_2+1} . At each pruning step, tokens are deleted so that after the second module, k hidden states $\hat{\mathbf{h}}_{l_2}$ are outputted. In **Resid** we use two distinct hidden states from separate transformer layers, combined by summing h_{l_1} and h_{l_2} , and the result is input into the scorer network $\mathcal{S}_\phi$. In **RANDOM** we include a baseline variant where tokens are randomly selected. If performance falls below this naive method, it indicates inadequate functionality.

We selected $l_1 = 20$ and $l_2 = 25$ because, as shown in Figure 2.6, these layers produce the most informative hidden states for token pruning. Table 2.8 presents the results of the architectural variants. MISTRAL-PP achieves the best and most consistent performance across tasks, with ATTN and RESID showing competitive results in specific settings (i.e.,

TABLE 2.8. Results of architectural variants over MISTRAL-7B in the text classification tasks.

Model	BOOLQ		BEAVERTAILS		ANLI		IMDB	
	F1-M	F1-m	F1-M	F1-m	F1-M	F1-m	F1-M	F1-m
MISTRAL-PP	89.39	90.03	86.43	86.69	72.74	72.90	96.68	96.68
MISTRAL-PP-RANDOM	72.78	73.27	84.31	84.64	65.45	66.00	95.43	95.43
MISTRAL-PP-ATTN	89.46	89.97	85.60	86.03	72.21	72.30	96.56	96.56
MISTRAL-PP-ENS	89.10	89.79	85.80	86.09	71.47	71.60	96.44	96.44
MISTRAL-PP-PROG	88.04	88.69	85.68	86.03	71.65	72.00	96.48	96.48
MISTRAL-PP-RESID	88.56	89.17	85.78	86.09	72.67	73.10	96.13	96.12

BOOLQ and ANLI), suggesting that attention scores and multi-layer representations can partly substitute the scorer network. In contrast, ENS and PROG do not provide clear benefits, while the strong gap with the RANDOM baseline confirms that our pruning strategies capture meaningful information.

2.5.6 Conclusion

We introduce PRUNEPERT, an approach that augments PLMs with a token-pruning layer to select essential input tokens for prediction. Quantitative and qualitative analyses on summarization tasks show that our method performs comparably to a state-of-the-art linear transformer. Notably, PRUNEPERT is more efficient, achieving an average speed-up of nearly 2x. Additionally, in four classification tasks, we demonstrate that it enhances LLM accuracy, improving the F1 score by up to 2 percentage points. It also outperforms state-of-the-art token pruning methods (LLMLINGUA, LLMLINGUA-2) in both efficiency—since it does not require a preprocessing step—and effectiveness, underscoring the benefits of learning token importance in an end-to-end fashion. We also show that in text summarization, the selected tokens are predominantly nouns and verbs, whereas in classification tasks, a higher proportion of punctuation and stopwords are selected. Future work, following approaches proposed for communication networks [Lodi et al. \[2010a\]](#), [Moro and Monti \[2012a\]](#), could explore the tracking and propagation of knowledge refinements across sentences.

Ethics Statement

While PLMs improve various NLP tasks like text summarization and classification, they have limitations in maintaining accuracy and fidelity in generative tasks. Therefore, we recommend that outputs from our models and other generative models be reviewed by domain experts before use. This ethical safeguard is crucial to prevent the spread of incorrect or misleading information, especially in clinical and scientific contexts where accuracy and reliability are critical.

Chapter 3

Knowledge Distillation

3.1 Definition and Taxonomy of Knowledge Distillation

Knowledge Distillation (KD) refers to the technique of transferring knowledge from a large, high-capacity model (the teacher) to a smaller, more efficient model (the student). It was first explored as a form of model compression by [Bucila et al. \[2006\]](#), who showed an ensemble’s knowledge could be condensed into a single model, and later formalized by [Hinton et al. \[2015\]](#) as “distilling” a teacher model’s knowledge into a compact student. In KD, the student is trained under the supervision of the teacher, typically by mimicking the teacher’s outputs or behavior on a training dataset. The student model thus learns to reproduce the function learned by the larger model, achieving similar performance with far fewer parameters. KD has become a cornerstone for creating efficient models deployable on resource-constrained devices without significant loss in accuracy. It is widely applied in both computer vision and NLP, especially as modern deep networks and Transformers have grown to billions of parameters, posing challenges for deployment [Mansourian et al. \[2025\]](#).

Taxonomy of Knowledge Distillation Over the years, various forms of knowledge distillation have been developed. The knowledge that a teacher can distill into a student generally falls into three broad categories:

- **Response-Based Knowledge:** Also called logit-based distillation, this is the traditional form of KD introduced by [Hinton et al. \[2015\]](#). It transfers the teacher’s output probabilities (logits) as “soft targets” for the student. Instead of training the student only on hard labels, the student is trained to match the teacher’s predicted probability distribution over classes, often using a softened softmax (controlled by a temperature) to reveal richer information (so-called “dark knowledge”) in the teacher’s outputs. This response-based approach is simple and broadly effective, and it was popularized due to its success in improving student performance on classification tasks.
- **Feature-Based Knowledge:** These methods transfer intermediate representations from the teacher’s hidden layers to the student. The student is guided to

align its feature maps or activations with those of the teacher, usually by minimizing distance between teacher and student feature vectors at certain layers. The concept of feature distillation began with FitNets [Romero et al. \[2015\]](#), which introduced using a teacher’s hidden layer as a “hint” for a student network. By imitating not just outputs but the teacher’s learned features, the student can capture more fine-grained or hierarchical knowledge. Feature-based KD has gained popularity for conveying richer, more nuanced information than logits alone. For example, aligning activation maps or attention matrices between teacher and student networks.

- **Relation-Based Knowledge:** Also known as similarity-based distillation, this category has the student learn the relationships or structural information that the teacher has learned about the data. Instead of matching individual outputs or features, the student is trained to match higher-order relationships such as pairwise similarities or distances between examples in the teacher’s representation space. For instance, the Flow of Solution Procedure method [Yim et al. \[2017\]](#) had the student mimic the inner-product of feature pairs between two teacher layers. Similarly, other relation-based approaches transfer metrics like the distance or angle between feature vectors, or the relative rank of logits, as a way to inject the teacher’s structural knowledge. This type of distillation is particularly useful in tasks where understanding the relationship between data points is crucial (e.g. metric learning, ranking, or graph-based problems). By learning these interdependencies, the student gains a form of generalized insight beyond individual predictions.

Apart from the source of knowledge, KD methods are also classified by distillation scheme or timing:

- **Offline Distillation:** The classic scheme where a teacher model is first pre-trained (often on large-scale data), and then a separate student model is trained under the teacher’s guidance in a subsequent stage. The teacher’s parameters remain fixed during student training. This is the most common setting, for example, using a pre-trained BERT [Devlin et al. \[2019a\]](#) as teacher to train a DistilBERT [Sanh et al. \[2019\]](#) student on the same task is offline KD. Offline distillation is straightforward and effective when a high-quality teacher is available.
- **Online Distillation:** In this scheme, there is no fixed pre-trained teacher; instead, the teacher and student (or multiple students) are trained simultaneously, each guiding the other. One approach is mutual learning [Zhang et al. \[2018\]](#), where two networks learn from each other’s predictions in tandem, effectively each one acting as a teacher for the other at each step. Alternatively, a cohort of student models can learn together and distill a consensus knowledge. Online KD can be beneficial when no single large teacher is available, though it requires careful coordination of training.
- **Self-Distillation:** A special case of online KD is self-distillation, where the teacher and student are actually the same model (or share the same architecture). Here, a

network teaches itself by internalizing knowledge from its own predictions or from deeper layers to shallower layers. For example, born-again networks [Furlanello et al. \[2018\]](#) train a model, then treat that same model (or its earlier generation) as the teacher for a new instance of the same model, iteratively improving performance. Self-distillation blurs the line between teacher and student, there is no larger teacher model; instead the model refines its knowledge by learning from its own outputs, often akin to a form of model ensemble or smoothing over training epochs. Interestingly, self-distillation has been theoretically linked to phenomena like label smoothing and regularization.

Further categorization can be based on the number of teachers and the modalities or domains. A student may learn from an ensemble of multiple teachers (multi-teacher distillation), or a teacher might be of a different modality or tasked with a different objective (cross-modal or cross-task distillation). In multi-teacher distillation, the student aggregates knowledge from several teacher models, which could provide complementary information or diverse views. This setup, also called knowledge amalgamation [Shen et al. \[2019\]](#), can enhance the student’s generalization by exposing it to a broader range of knowledge than any single teacher. However, it also introduces challenges in balancing and fusing the multiple sources. Cross-modal KD, on the other hand, refers to distilling knowledge from one data modality to a model of another modality. For example, using a vision model’s knowledge to improve an NLP model or vice versa, or distilling multilingual knowledge from one language to another. These variants extend the flexibility of KD beyond the standard single-teacher single-student paradigm.

In summary, knowledge distillation encompasses a spectrum of techniques for model compression and knowledge transfer. It can be taxonomized by what knowledge is transferred (outputs vs. features vs. relations), when and how the transfer happens (offline, online, self, etc.), and the configuration of teacher/student models (one-to-one vs. many-to-one, same architecture vs. different, same modality vs. cross-modal). This taxonomy provides a framework to understand the diverse methods under the KD umbrella.

3.2 Challenges in Knowledge Distillation

Despite its success, knowledge distillation involves several challenges that researchers and practitioners must navigate to effectively transfer knowledge from teacher to student.

Identifying What Knowledge to Distill A fundamental challenge is deciding which information from the teacher will be most useful to the student i.e. the source of knowledge. Using the teacher’s output logits is the most straightforward approach, but it may not always be sufficient or even feasible. For instance, if the task has no class labels or the model’s goal is to produce embeddings (as in retrieval or representation learning), plain logit-based distillation is not applicable or informative. In large pre-trained models, much of the valuable knowledge resides in learned feature representations and

relationships, not just final outputs. Thus, feature-based and similarity-based distillation become crucial, yet selecting which layers or features to transfer is non-trivial. The teacher may have many layers and channels and aligning the student to the wrong layer or mismatched feature can hinder learning. Choosing appropriate intermediate layers with “similar semantic information” between teacher and student is important for success in feature distillation. This challenge is exacerbated when teacher and student architectures differ. The student might not even have a layer corresponding to the teacher’s layer, a scenario requiring careful design of distillation objectives. In summary, extracting the right form of knowledge (logits vs. features vs. relational cues) and from the right parts of the network is a key difficulty.

Limited Access to Teacher’s Knowledge In some cases, especially with the rise of proprietary large language models, the full teacher model is not accessible to the distiller. The teacher might be a black-box API that only provides output predictions, or even just final answers in the case of an LLM, but not internal representations or logits. For example, recent state-of-the-art LLMs like GPT-4 [Achiam et al. \[2023\]](#) are not open-source, so one cannot extract their intermediate layer features or logits for feature-based distillation. This limits the distillation techniques to what can be gleaned from the teacher’s outputs alone. Black-box distillation is challenging, the student must learn from only the teacher’s final answers or probabilities, which may not convey all the reasoning the teacher used. Some recent strategies, such as chain-of-thought (CoT) distillation, try to address this by having the teacher generate intermediate reasoning steps (e.g. explanations) that the student can learn from, rather than forcing the student to mimic the teacher’s hidden layers. Nonetheless, when access to the teacher’s internals is restricted, KD becomes more difficult and sometimes less effective.

Choosing the Right Distillation Scheme Deciding on an appropriate training scheme for KD is another challenge. If a high-quality pre-trained teacher is available, offline distillation is typically effective. But in scenarios where no such teacher exists, one might attempt online distillation or self-distillation. Each scheme has trade-offs, for instance, online mutual distillation can sometimes lead to peer models converging to a suboptimal agreement if not carefully managed. Using multiple teachers (e.g. an ensemble of teachers) is promising because it can provide diverse knowledge, but it introduces complexity in training and high computational cost (multiple large models must be run in parallel). Multi-teacher KD demands strategies for knowledge aggregation, how to combine potentially conflicting signals from different teachers. Without careful weighting or fusion mechanisms, the student might be overwhelmed or learn an incoherent mix of knowledge. Researchers have explored treating multi-teacher distillation as a mixture-of-experts problem, or using meta-learning to adaptively weight each teacher’s contribution, but these add complexity and runtime cost. In practice, most works default to the simpler offline single-teacher setup due to these difficulties [Mansourian et al. \[2025\]](#).

Capacity Gap Between Teacher and Student Perhaps the most cited challenge is the capacity gap, which arises when the teacher network is significantly larger and more expressive than the student. Intuitively, one might think a bigger teacher always helps and indeed a sufficiently larger teacher can provide richer knowledge, but if the student is too small to absorb that knowledge, training can stall or the student ends up underperforming relative to a scenario with a moderately-sized teacher. [Cho and Hariharan \[2019\]](#) famously observed that a too large teacher can actually hurt distillation, as the student struggles to mimic knowledge that exceeds its representational capacity. The teacher’s learned function might be too complex, and the student cannot approximate it well, leading to poor convergence (this is related to differences in the “receptive fields” or complexity of functions each can represent. The capacity gap is even more pronounced when distilling from gigantic foundation models or LLMs into tiny models, where the student sees an immense knowledge gap. Bridging the capacity gap is an active research area. Solutions include teacher-assistant distillation, where an intermediate-sized model is used as a bridge between a very large teacher and a small student. This staged distillation has been shown to improve student performance by easing the learning burden. Other techniques reduce the gap by making the teacher’s knowledge more student-friendly, for example, simplifying the teacher’s outputs or focusing on essential aspects of knowledge. Adapting the distillation loss (e.g. using higher temperature to soften the knowledge, or regularizing the student) can also help mitigate capacity mismatch. Nonetheless, the capacity gap remains a fundamental limitation. A too-small student simply cannot replicate a huge teacher’s full behavior, so practical distillation often finds a sweet spot in teacher-student size ratios.

Architectural Mismatch When the teacher and student have different architectures (a common scenario, e.g. teacher is a Transformer, student is a smaller Transformer or even a different model family), aligning knowledge becomes challenging. If the student network lacks layers or modules present in the teacher, it’s not obvious how to perform feature distillation e.g., which student layer should imitate a particular teacher layer? Architectural differences can lead to semantic misalignment, where a given internal feature in the teacher does not have a clear counterpart in the student [Cho and Hariharan \[2019\]](#). This often leads to degraded performance if not handled carefully. Approaches to tackle this include mapping teacher layers to student layers based on some criterion (like matching spatial resolution or activation types), using projection layers to transform teacher features to student’s dimensionality, or focusing on more abstract forms of knowledge (like relational knowledge) that are less architecture-dependent. For example, if a teacher is a big Transformer with dozens of attention layers and the student has far fewer, one might distill only the final representations or attention distributions in aggregate, rather than trying to match layer-by-layer. Some research suggests distilling behaviors (e.g. teacher’s reasoning traces or intermediate outputs like chain-of-thought) instead of raw features to bypass architectural mismatch. Ultimately, a student with a different inductive bias from the teacher will never exactly mirror the teacher, and how to best transfer knowledge across architectures remains a challenge.

Data Requirements and Domain Shift KD typically assumes the availability of a dataset (unlabeled or labeled) on which the teacher’s knowledge is distilled to the student. If the original training data for the teacher is not available (due to privacy or size), or if we want to distill knowledge to a new domain where labeled data is scarce, it can be problematic. In scenarios with limited data, the student might overfit to the small distillation set or fail to generalize the teacher’s knowledge broadly. A promising line of work is data augmentation or synthetic data generation for KD. If real data is scarce or sensitive, one can use the teacher model itself to generate synthetic examples for the student. This is known as data-free knowledge distillation, and it addresses both data scarcity and privacy concerns. By generating artificial data, for instance, using a generative model or the teacher model in generative mode, one can distill without exposing or requiring the original dataset. However, ensuring the synthetic data is representative enough of the tasks is challenging, and there is a risk of the student learning artifacts of the generation process rather than true knowledge.

Optimization and Stability Issues Training a student to mimic a teacher introduces additional hyperparameters and objectives that require balance. For example, when using both ground-truth labels and teacher soft targets, one must choose the relative weight of the distillation loss vs. the standard loss, as well as an appropriate temperature for softening logits (as too high or too low temperatures can lead to ineffective distillation). If the teacher’s outputs have errors or biases, the student might inadvertently learn these (error propagation). Moreover, in some cases, the student might focus too much on mimicking the teacher and not enough on the true task, leading to degraded performance if the teacher is not perfect. Ideally, the teacher should be high-performing, a low-accuracy teacher can limit the student’s ceiling. From an optimization perspective, adding the distillation loss can change the loss landscape, sometimes for the better (as a regularizer or guidance), but it can also introduce shallow minima where the student just superficially matches the teacher without truly learning the task. Practitioners have found that careful training schedules e.g., first train student on teacher signals, then fine-tune on true labels, or vice versa, can help.

Privacy and Legal Concerns While KD can be used to preserve privacy (by using teacher’s knowledge in lieu of raw data), it can also raise legal/ethical concerns. If the teacher is a proprietary model or trained on copyrighted data, is the distilled student considered a derivative work? Recent commentary notes that using a commercial API-based model as a teacher might violate terms of service or intellectual property rights if done without permission. This hasn’t been fully resolved in policy, but it’s a practical consideration: many cutting-edge teachers (like commercial LLMs) come with restrictions on use. Researchers must be mindful of these constraints when choosing a teacher for distillation.

In summary, knowledge distillation, while powerful, is not a trivial plug-and-play technique. One must address what knowledge to transfer, how to transfer it, and under what conditions, all while coping with differences in model capacity and architecture.

Ongoing research is actively tackling these challenges, devising more robust distillation algorithms and adaptive strategies to make KD effective even in difficult settings (e.g. huge teacher vs tiny student, no original data, etc.). Many of these challenges also point to future research directions, such as distilling reasoning abilities of LLMs, distilling under distribution shifts, and combining KD with complementary compression methods like pruning and quantization.

3.3 Knowledge Distillation Methodologies

Building on the definitions and challenges above, we now delve into the methodologies of knowledge distillation, the concrete techniques and strategies that have been developed to transfer knowledge from teacher to student. We structure this discussion by the forms of knowledge being distilled and by notable variations in the distillation process, highlighting key methods from the literature.

3.3.1 Response-Based Distillation

Response-based distillation is the foundational methodology introduced by [Hinton et al. \[2015\]](#). In this approach, the student learns directly from the teacher’s output layer. The teacher’s prediction, often a probability distribution over classes, is used as a soft target for the student network. The classic implementation uses a softmax temperature: the teacher’s logits z_i (for class i) are converted into probabilities $p_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$ with a temperature $T > 1$ to soften the distribution (higher T produces a softer, more uniform probability vector that reveals less confident class relationships). The student is trained to produce a similar output distribution by minimizing the Kullback-Leibler divergence (or another divergence) between the teacher and student probabilities. At the same time, the student may also be trained on the true hard labels (ground truth) with the usual loss. The overall objective is a weighted sum of the standard loss and the distillation loss. The intuition behind response-based KD is that the teacher’s output probabilities encode rich information about the task that goes beyond the hard label. For example, if the teacher is classifying an input and assigns a 0.60 probability to the correct class A and 0.30 to B and 0.10 to C, the relative magnitudes (A is only somewhat more likely than B) tell the student that B is the second-best guess and not far off. A student trained on hard labels would only know A is correct and nothing about B or C. By imitating the teacher’s soft output, the student can learn not just the primary classification but also the relative confidence and confusions of the teacher, which acts as a form of regularization and guidance. This often leads to improved generalization, as the student is encouraged to model the full output distribution. Key works in this methodology include [Hinton et al. \[2015\]](#), which demonstrated significant compression with minimal loss in accuracy on image classification. [Ba and Caruana \[2014\]](#) also independently showed that a deep network’s function could be transferred to a shallow network by matching logits. In NLP, an early adoption was seen in compressing large neural machine translation models and language models. For example, DistilBERT [Sanh et al. \[2019\]](#) leveraged a form of soft-

target distillation to compress BERT, and sequence-level knowledge distillation [Kim and Rush \[2016\]](#) applied the idea to teacher and student sequence models by having the teacher generate probability distributions over output sequences. One important consideration in response-based KD is setting the temperature properly. A higher T produces softer targets that place more weight on secondary classes, which can be helpful for student learning early on, but if T is too high, the targets may become too flat and uninformative. Typically T is chosen empirically (e.g. 2, 5, 10). The weight between the distillation loss and ground-truth loss is another hyperparameter. If the teacher is very accurate, one might rely more on its soft targets; if the teacher has errors, the student should not follow it blindly, so the ground truth loss remains important. Overall, response-based distillation remains a widely used and effective methodology due to its simplicity. It does not require structural alignment between teacher and student, only the outputs need to be comparable. This makes it applicable even when the student is architecturally very different or much smaller. However, as noted earlier, solely relying on logits might miss internal reasoning, hence advanced methods often extend beyond pure response imitation.

3.3.2 Feature-Based Distillation

Feature-based distillation methods aim to transfer knowledge from the intermediate layers of the teacher to the student. The rationale is that the teacher’s hidden layers capture valuable representations, for example, in a Transformer model, early layers might encode syntactic information and later layers semantic information. If the student can learn these intermediate representations, it stands to benefit by building a similar hierarchy of understanding. A seminal work in this category is FitNets [Romero et al. \[2015\]](#), which introduced the idea of using a teacher’s hidden layer output as a “hint” for the student. In FitNets, the authors select one of the teacher’s mid-level layers and force the student to produce a matching feature map at one of its own layers, often a deeper layer in the student, since the student is usually shallower. They add a regression loss on the feature activations, after projecting to the same dimension if needed, such that the student’s intermediate feature F_s approximates the teacher’s feature F_t . Training is done in a staged way. First, the student trains to match the hint layer, then it continues training on the primary task with possible further distillation on outputs. This approach showed that training with intermediate feature alignment can significantly boost student performance, especially for deeper student networks that otherwise are hard to train from scratch. Following FitNets, many feature-based KD techniques emerged. [Zagoruyko and Komodakis \[2017\]](#) proposed transferring attention maps, instead of raw features, have the student mimic where the teacher’s network focuses its attention (spatial or channel-wise attention). This attention transfer method computed attention as a sum of absolute feature values along certain dimensions and made the student match those attention distributions. The idea is that if the student learns to focus on relevant parts of the input similar to the teacher, it will perform better. Other methods looked at deeper layer-by-layer distillation. For example, activation-based and affinity-based losses have been used. The student can be trained with loss terms that measure the distance be-

tween its layer outputs and the teacher’s outputs for multiple layers (possibly all layers, if architectures align). Techniques like employing an additional convolution to transform student features before comparing (to account for dimension mismatch) are common. In NLP, feature distillation has been used for compressing large Transformers. For instance, TinyBERT [Jiao et al.](#) distilled BERT into a much smaller model by matching not only logits but also embedding layer outputs and the Transformer attention matrices and value matrices at multiple layers. They used a combination of losses on these internal representations, which led to a student that more faithfully imitated BERT’s behavior on a variety of tasks compared to only output distillation. Feature-based KD tends to yield a student that is closer in representation space to the teacher. This often translates to better performance on tasks that rely on internal feature quality. If the student will be fine-tuned further or used for transfer learning, having similar embeddings as the teacher is beneficial. As mentioned earlier, a challenge is selecting the right layers. If a teacher is much deeper, one might either skip some teacher layers or have the student try to absorb multiple teacher layers into one of its layers. There’s also a question of how to weight multiple feature losses. Some methods introduce learnable weights for each layer’s distillation loss or use an adaptive weighting scheme. In summary, feature-based methodologies enrich the student’s training signal by providing fine-grained guidance at multiple levels of abstraction. This often leads to better convergence and higher accuracy, especially in tasks where output alone doesn’t tell the full story (like embedding learning). However, these methods often require more effort in implementation (access to teacher’s internals, aligning dimensions) and may not always be applicable if teacher and student are too different, though one can distill from a Transformer teacher into, say, an LSTM student by intermediate representation matching in a latent space, with proper care.

3.3.3 Relation-Based and Similarity-Based Distillation

Relation-based distillation extends beyond individual activations to capture relationships and structure in the data as learned by the teacher. These methods recognize that a knowledgeable teacher not only produces good outputs and features, but also organizes information in a meaningful way, for instance, clustering similar inputs in the feature space, or maintaining certain relational knowledge among examples, such as ordering by difficulty, or semantic similarities. A classic example is the Relational Knowledge Distillation (RKD) [Park et al. \[2019\]](#), which had the student learn distances and angles between all pairs of data points as embedded by the teacher. Concretely, if $d_{ij}^t = |F_t(x_i) - F_t(x_j)|$ is the distance between teacher features for inputs i and j , and θ_{ijk}^t is the angle between teacher’s vectors for i, j, k , the student would be penalized for differences between its corresponding distances d_{ij}^s and angles θ_{ijk}^s . By doing so, the student attempts to reconstruct the geometry of the teacher’s feature space. Even if the absolute features differ, as long as the relational structure is preserved, the student can be said to have learned the teacher’s representation manifold. Earlier, we mentioned the FSP method [Yim et al. \[2017\]](#), which is a relation-based approach focusing on layer-wise relations. It considered relationships between features from two layers in the teacher,

essentially encoding how one layer’s output transforms to the next, and made the student mimic that transformation. The general idea is that higher-order knowledge, like “these 10 images form a cluster in teacher’s feature space, distinct from another cluster of 10 images”, can be very informative. It’s a way to transfer the teacher’s conceptual organization of the data. In NLP, one could imagine relation-based KD being used for tasks like teacher knowing which sentences are semantically similar. The student could learn a similar similarity function. One benefit of similarity-based distillation is that it can sometimes be applied even when teacher and student representations live in different dimensional spaces or when direct feature alignment is hard, because relationships can be computed in any space, as long as you can compute a distance or similarity). It also tends to emphasize generalization. By focusing on relations, the student is not forced to match precise values, but rather the structure, which might be more robust to differences. However, relation-based KD often complements, rather than replaces, direct feature KD. Many works combine a relation loss with a standard distillation loss. There is also a computational consideration. Computing pairwise relations among n samples can be $O(n^2)$ or $O(n^3)$ in some cases (like if triple-wise relations are used), which can be heavy. Typically, one uses mini-batches and computes relations within batches. In practice, methods like Contrastive Representation Distillation (CRD) [Tian et al. \[2019\]](#) have shown strong results. CRD uses a contrastive learning objective where the student must identify the matching teacher feature for a sample among a set of alternatives, effectively distilling via a contrastive loss that preserves instance-wise similarity. This can be seen as a form of relational KD. The relationship of “this belongs to this sample vs. others” is being transferred. Overall, relation-based methodologies enrich KD by transferring structural knowledge, the “grammar” of the learned representation, beyond individual data points. This is particularly useful for tasks where the relative understanding is key (rankings, metric learning, etc.), and it has shown improvements even in classification tasks when combined properly with other losses.

3.3.4 Self-Distillation and Online Distillation Techniques

While the above categories usually assume a fixed teacher, self-distillation flips the script by having the model distill knowledge into itself. In self-distillation, there is no separate, larger teacher model; instead, the training procedure is designed such that a model effectively teaches itself. [Furlanello et al. \[2018\]](#) introduced Born-Again Networks (BAN), where one takes a neural network, trains it, then treats that trained network as a teacher to initialize or supervise a new student of the same architecture. Repeating this process for several generations sometimes yields a student (the “born-again” network) that outperforms the original network. It’s as though training a network with itself multiple times can escape local minima or find a more refined function. Each subsequent generation’s knowledge is a distilled version of the previous, potentially acting like an ensemble through time. Some methods incorporate self-distillation during a single training run by adding auxiliary losses between different layers of the same network. For example, a deeper layer’s output might serve as a “teacher” for an earlier layer’s output in the same network. The network is encouraged to have early layers already produce good

representations (as if they were trying to match the final layer). This acts as a form of regularization, each layer is learning from the eventual “teacher” (the final layer) gradually. As a form of online distillation, one can train a cohort of models simultaneously, each one learning from others. [Zhang et al. \[2018\]](#) showed that two networks of the same size trained from scratch can serve as each other’s teacher – they minimize the divergence between their predictions throughout training. This surprisingly led to better performance than training them independently (and even sometimes better than using one as a fixed teacher). The process can be extended to multiple students all distilling from each other (sometimes called cooperative learning). There are also approaches where a large teacher is distilled online, for example, gradually transferring knowledge as the student trains, instead of after full teacher training (though usually the teacher is fixed to ensure stability). From a practical point of view, self-distillation is attractive because it does not require a separate teacher model at training time (which might be expensive to run). Every model can potentially benefit from KD techniques even if you only train a single model. For example, recent studies in vision and NLP have integrated self-distillation into the training objective of one model to get gains as if a teacher was present. However, self-distillation doesn’t always guarantee improvement – if not carefully configured, a model might simply learn to predict its own initial predictions and get stuck. Also, mutual learning can converge to a agreement that’s not optimal if both models have the same blind spots.

3.3.5 Multi-Teacher and Ensemble Distillation

When one teacher is good, multiple teachers could be better – that’s the intuition behind multi-teacher distillation. Instead of a single teacher model, the student is given guidance from an ensemble or a set of teacher models, which may be trained on different data, have different architectures, or specialize in different tasks. By consolidating knowledge from several experts, the student could in theory achieve broader or more robust understanding. There are a couple of ways multi-teacher KD is implemented: Ensemble Logit Distillation: The simplest approach is to have multiple teachers each provide an output distribution, and then either average these distributions or use a weighted average as the target for the student. Essentially, the student distills the ensemble’s output. This was actually one of the original motivations: compressing an ensemble of many models into one. The ensemble’s prediction is often more accurate and smooth than any single teacher, so the student can sometimes even outperform individual teachers by imitating the consensus. However, one must be careful if the teachers disagree significantly; in such cases, averaging might produce a target distribution that no single teacher had, and the student learning process might become difficult. Sequential Distillation or Stage-wise: Another way is to sequentially distill from multiple teachers. For example, if you have Teacher A and Teacher B, you could first distill A into an intermediate student, then use that student (or combine with B) to distill into the final student. This is related to the teacher-assistant idea where one of the teachers could be seen as an assistant simplifying knowledge from a more complex teacher for the student. Or each teacher might handle a subset of the task (like teacher1 for task1, teacher2 for task2), and you distill

all into one multi-task student. **Knowledge Amalgamation:** A term used when the goal is to fuse knowledge from teachers that might have expertise in different domains into one student. For example, one teacher might be good at classifying animals, another at classifying vehicles; an amalgamated student could handle both if distilled properly. This often requires a careful design so that the student’s capacity is enough and that the training data covers both domains. The benefit of multi-teacher distillation is a potentially richer training signal. Each teacher might offer unique insights or correct different mistakes. The student could learn a more generalized solution. Indeed, studies have shown students from multiple teachers can achieve better generalization and accuracy than from a single teacher. The challenges include how to effectively merge the knowledge. If teacher outputs conflict, the student loss might be confused. Some works introduce an adaptive weighting: e.g., during training, measure which teacher’s predictions the student currently imitates best and weight that teacher less or more to balance. Another approach is to give each teacher’s output to the student as separate distillation objectives (like multi-objective optimization). There are also methods using gating – train a small gating network that decides which teacher’s knowledge to apply for each instance (like a student that dynamically attends to the appropriate teacher). All these add complexity. Additionally, multi-teacher KD is computationally costly: you have to run multiple forward passes (one per teacher) for each training sample, unless teachers’ predictions are precomputed. In a large-scale scenario, this could be prohibitive in terms of memory and speed. Therefore, multi-teacher is sometimes done in an offline manner where teachers’ outputs on the training set are saved first (if the training set is not huge). Despite these issues, multi-teacher distillation has been successfully used. An example in NLP is distilling knowledge from several monolingual models into one multilingual student or vice versa, or combining an ensemble of question-answering models into one. The student often ends up with diverse learned knowledge and improved robustness. Surveys summarize that multi-teacher provides the student with a “broader range of diverse and enriched information,” enhancing generalization, but requires careful handling of the capacity gap and knowledge aggregation.

3.3.6 Adversarial and Other Advanced Distillation Techniques

Beyond the main categories above, there are a number of advanced methodologies worth noting: **Adversarial Distillation:** Borrowing ideas from GANs, some techniques set up an adversarial game where the student tries to generate outputs (or features) that are indistinguishable from the teacher’s, as judged by a discriminator. For example, an adversarial distillation method might have a discriminator network that takes either teacher or student feature maps and must tell which is which, and the student is trained to fool the discriminator. This can encourage the student to match the teacher’s distribution of features more holistically. Adversarial KD has been applied in data-free scenarios too (Chen et al. 2019) where a generator produces dummy inputs that make the teacher and student output distributions close, using a GAN loss. The adversarial approach can capture higher-order similarities and is quite flexible, though training GAN-like setups can be tricky and unstable. **Adaptive Distillation:** Methods that adjust the distilla-

tion process on the fly. For instance, not all teacher knowledge might be equally useful throughout training – early on, the student might benefit from focusing on coarse knowledge, later on fine-grained. Some adaptive methods change the weight of the distillation loss over epochs, or change which layers to distill from as training progresses. Another adaptive idea is curriculum distillation: start distilling easier aspects of the teacher’s knowledge (maybe on easier examples or higher-level outputs) then gradually move to harder aspects. This ties in with the notion of teacher assistant which we discussed as bridging capacity gap – that is a form of adapting the teacher complexity to the student.

Cross-Modal Distillation: Although more relevant to multi-modal research, it’s interesting in an NLP context too. For example, one could use a strong image-captioning model (vision-language) as a teacher to improve a pure language model on descriptive text generation, by transferring some of the vision-grounded knowledge. Or distill a speech recognition model’s knowledge into a text model that predicts phonetic sequences. These scenarios expand KD beyond identical tasks, essentially using KD for transfer learning.

In NLP, a related idea is cross-lingual distillation: a high-resource language model or classifier (say English BERT for sentiment) can teach a low-resource language model (say for Spanish sentiment) by translating or using a multilingual teacher. The teacher’s outputs on Spanish data (even if teacher is an English model, maybe via a multilingual model) serve as supervision for the Spanish student.

Lifelong and Online Learning with KD: In continual learning, KD has been used to retain knowledge. When a model is learning new tasks sequentially, it can treat the older version of itself as a teacher to avoid catastrophic forgetting (this is essentially self-distillation across time). By preserving the previous outputs via a distillation loss while learning new data, the model doesn’t deviate too far on what it previously knew. This is common in continual learning algorithms (e.g., LwF – Learning without Forgetting, used KD to keep new model’s predictions close to old model’s for old tasks).

Knowledge Distillation in Large Language Models: Since the context of the thesis is large language models (LLMs), it’s worth noting how KD is specifically applied there. LLM Distillation has become a hot topic recently as researchers try to compress models like GPT-3 (175B parameters) or others into more manageable sizes. Techniques include output distillation on next-word prediction across vast corpora (to train a smaller language model to mimic a large one’s next-word distribution), as well as distilling specialized abilities (like reasoning or following instructions) from an LLM into a smaller model. For example, using ChatGPT or GPT-4 as a teacher to create a high-quality dataset of question-answer pairs, and then fine-tuning a smaller model on that – effectively performing KD.

3.3.7 Data-Free Knowledge Distillation and Synthetic Data Generation

As mentioned in the challenges, an exciting development in KD methodologies is data-free distillation – where the distillation process does not rely on the original training data (which might be unavailable or insufficient). Instead, synthetic or unlabeled data is used, often leveraged from the teacher itself. In computer vision, data-free KD has involved generating images by inverting the teacher model (producing inputs that maximize cer-

tain activation patterns) or using a generator network that produces pseudo-inputs on which teacher and student are trained in tandem. In NLP, data generation is more straightforward in a sense, because a large language model can directly generate text. Synthetic data generation for KD in NLP has gained traction due to the impressive generative capabilities of large models. Rather than manually annotating data, one can prompt an LLM to create a corpus of questions, answers, translations, etc., and then use those as training data for a smaller model. This effectively transfers knowledge: the teacher’s knowledge about language and the task is embedded in the synthetic examples it produces, and the student. Nonetheless, synthetic data generation as a KD methodology is extremely powerful in NLP because text is easy for large models to produce in unlimited quantities. The limitation becomes quality, not quantity. Ensuring the generated data is representative and challenging enough for the student is key. Techniques like filtering generated data, using multiple prompts, human in the loop to check some outputs, or gradually refining the generation via feedback can improve results. In summary, data-free and synthetic data distillation augment the traditional KD toolbox by removing dependence on original datasets. They leverage the teacher as not just a source of outputs on existing inputs, but as a source of new inputs and outputs. Ace-Attorney’s success in legal QA is a testament to the potential of this approach in specialized domains. We can expect to see more “generate-then-distill” frameworks, especially as LLMs continue to advance and the demand to deploy smaller specialized models grows.

3.4

ACE-ATTORNEY: Enhancing Legal Question Answering with Data Generation and Knowledge Distillation from Large Language Models

Originally published as: *Paolo Italiani, Gianluca Moro, and Luca Ragazzi. Enhancing legal question answering with data generation and knowledge distillation from large language models: P. Italiani et al. Artificial Intelligence and Law, pages 1–26, 2025*

Abstract

Legal question answering (LQA) relies on supervised methods to automatically handle law-related queries. These solutions require a substantial 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 thesis presents ACE-ATTORNEY, an LLM distillation approach devised to develop LQA data and supervised models without

human annotation. Given a textual prompt, a frozen LLM generates artificial examples that are used as knowledge to train a student SLM with an order of magnitude fewer parameters. Taking into account a realistic retrieval-based scenario to fetch the correct document for answer generation, we propose Selective Generative Paradigm, a novel approach designed to improve retrieval efficacy. Extensive experiments demonstrate the effectiveness and efficiency of distilled models on SYN-LEQA, our human-free synthetic dataset, and a public expert-annotated corpus. Notably, by using only a few dozen training samples, our best SLM achieves LLM-comparable performance with $\approx 1200\%$ less CO₂ emissions.

3.4.1 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) could help quickly capture the main points of new cases [Martinez-Gil, 2023], similar to summarization systems [Ragazzi et al., 2024c]. 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). 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 small language models (SLMs) trained on synthetic data produced by LLMs [Meng et al., 2022, Ye et al., Gao et al.]. 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., 2023b]. However, previous work on KD has focused mainly on text classification, neglecting generative tasks such as QA. Furthermore, upon closer inspection, existing LQA datasets¹ exhibit at least one of the following limitations: (i) they concern specialized domains with constrained scope, such as tax law [Holzenberger et al.]; (ii) they are restricted to multiple choice [Zheng et al., Bongard et al.] or few-word replies [Sovrano et al.]; (iii) the answer is a verbatim extraction from the context [Ravichander et al., Ahmad et al.].

Motivated by the power of LLM text generation and the scarcity of LQA contributions, we present ACE-ATTORNEY (Figure ??), an LLM KD approach to automate the generation of LQA datasets and models. First, we use a frozen instruction-tuned LLM to create a comprehensive synthetic QA dataset via carefully designed prompts on legal corpora, filling the dearth of publicly available LQA datasets.² This phase is enhanced by a

¹As widely accepted by the community to track advance progress, we analyzed contributions for English.

²Note that our study aims to generate data ex novo, going beyond the purpose of data augmenta-

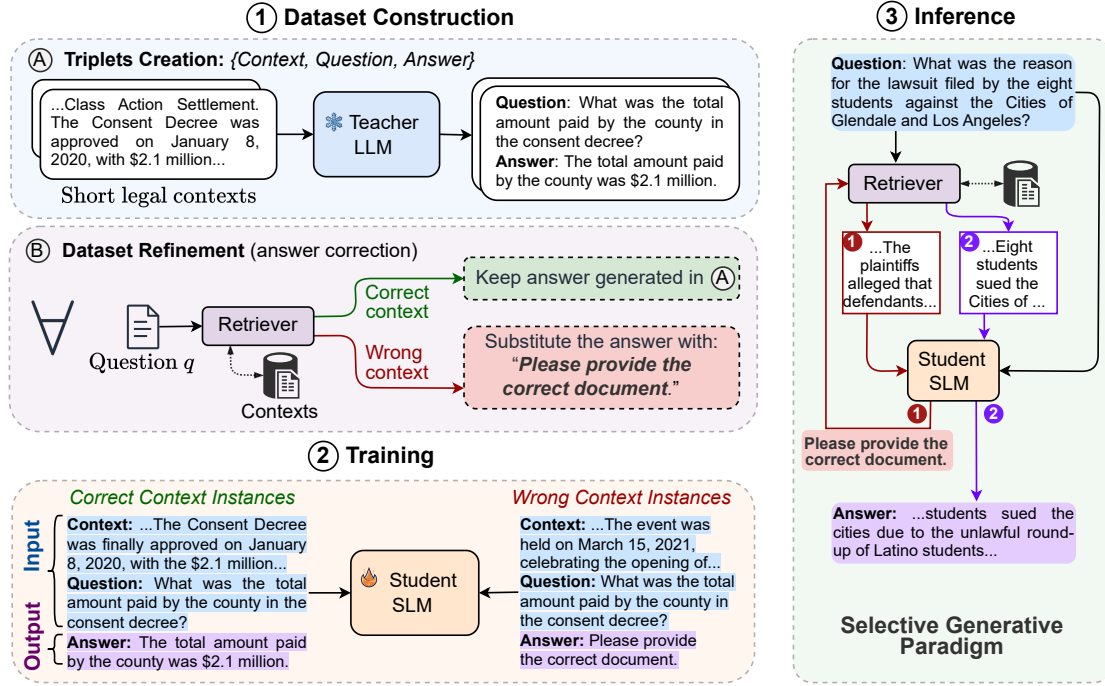


FIGURE 3.1. ACE-ATTORNEY overview. (1) A frozen LLM generates {context, question, answer} triplets from short legal contexts. For each question, we retrieve the corresponding context, manually adjusting the answer if the context is incorrect. (2) We then train an SLM on these triplets. (3) During inference, we employ a Selective Generative Paradigm, where the SLM provides an answer based on the relevance of the retrieved document.

data refinement process, where specific instances are modified to cover realistic scenarios in which an incorrect context is used to answer a given question. Subsequently, a student SLM is trained on the artificial question–answer pairs, operating in a pure zero-shot setting without involving human annotations throughout the entire pipeline. We based our analysis on a realistic scenario that involves retrieving the correct document within a cluster to answer a determined question. To this end, taking inspiration from existing works on relevance assessment [Faggioli et al., 2023, MacAvaney and Soldaini, 2023, Thomas et al., 2024], we present Selective Generative Paradigm (SGP), a methodology grounded in iterative refinement [Domeniconi et al., 2014, 2015b], specifically designed to improve the effectiveness of the retrieval module. Thanks to our prior dataset refinement process, the generative model—tasked with answering the question—can directly evaluate the pertinence of the retrieved document and request a new one if necessary. ACE-ATTORNEY accommodates models of different sizes without presumption on their architecture, fulfilling the available hardware infrastructure.

For experimentation, we consider privacy policies and civil rights lawsuits as the raw legal knowledge to build SYN-LEQA, a new synthetic LQA dataset originated by tion [Ghosh et al.].

LLM KD. We then fine-tune multiple million-scale SLMs on it and perform in- and cross-domain experiments on a real human-annotated LQA dataset.

In summary, our main contributions are the following:

- We explore synthetic data generation for legal applications, presenting ACE-ATTORNEY, a model-agnostic approach that operates via LLM KD to automate the production of data and small specialized models. With our approach, we create SYN-LEQA, the first high-quality synthetic dataset for generative LQA.
- We propose SGP, which allows the generator and the retriever to work jointly to select the correct document to answer the question.
- By benchmarking multiple neo-released billion-scale LLMs in LQA, we show that our distilled SLMs—despite having order-of-magnitude fewer parameters—outperform LLMs with $\approx 1200\%$ less CO₂ emissions, even obtaining LLM-comparable results in a real-case scenario of just a few dozen training samples.

3.4.2 Related Work

Legal Question Answering. LQA has been a long-standing challenge for legal intelligence applications [Kim et al., 2016], fueled by the overwhelming influx of new legal information. Consequently, to formulate and facilitate practical remedies, several datasets have been proposed and publicly released. CASEHOLD [Zheng et al.] collects more than 53K multiple-choice questions to identify the relevant holding of a cited case. PRIVACYQA [Ravichander et al.] aggregates 1750 questions on privacy policies of mobile applications. POLICYQA [Ahmad et al.] provides 714 human-annotated inquiries for a wide range of privacy practices. To the best of our knowledge, there are no open-access datasets that offer generative answers, although this is often a user requirement for lawyers [Zhang et al.].

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., 2021b]. 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. [2022] 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. Ye et al. [2022] push this concept to its extreme by training LSTMs on synthetic data. However, as far as we know, data generation has never been explored for legal NLP tasks,

TABLE 3.1. Dataset statistics for both SYN-LEQA and PRIVACYQA_{gen}. The word count is computed with the NLTK tokenizer [Bird et al.].

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

despite the benefit of bypassing the excessive annotation effort required by in-domain experts to create high-quality labeled examples.

3.4.3 Ace-Attorney

We introduce ACE-ATTORNEY, a new approach devised to develop LQA-specific data and supervised models through LLM KD.

Task Definition. LQA can be formally defined as follows. Let $\mathbf{D} = \{\mathbf{d}_1, \dots, \mathbf{d}_n\}$ 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: $\{\mathbf{d}'_1, \dots, \mathbf{d}'_n\} = \mathcal{R}(\mathbf{D} \mid \mathbf{q})$. Then, a generative model $\mathcal{G}$ produces the answer $\mathbf{a}$ using both $\mathbf{q}$ and $\mathbf{d}'_1$, namely $\mathbf{a} = \mathcal{G}(\mathbf{q}, \mathbf{d}'_1)$.

3.4.3.1 Syn-LeQA

Motivated by the inaccessibility of publicly available LQA datasets, this work focuses on the generation of examples $\mathcal{E} = \langle \mathbf{q}, \mathbf{d}'_1, \mathbf{a} \mid \mathbf{d}'_1 \in \mathbf{D} \rangle$, composing SYN-LEQA, our synthetic LQA dataset for training $\mathcal{G}$. Table 3.1 shows the statistics of our dataset, while Table 3.2 displays two random examples.

Challenges. This data generation process faces several challenges. **C1** The documents in $\mathbf{D}$ leveraged to generate examples $\mathcal{E}_1, \dots, \mathcal{E}_{|\mathcal{E}|}$ 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 $\mathbf{D}$, so that we can tackle **C1**. **(1) Civil rights lawsuits:** we consider short summaries of MULTI-LEXSUM [Shen et al., 2022], a legal corpus for multi-document summariza-

TABLE 3.2. Two examples taken from SYN-LEQA featuring the root document and the corresponding generated question-answer pair.

Document #1	
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 #1	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 #1	No, the court ruled in favor of the defendants and awarded no relief.
Document #2	
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 #2	What is one specific reason that Parsec Properties may suspend or terminate a user’s account and rights to use the Parsec Properties?
Answer #2	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.

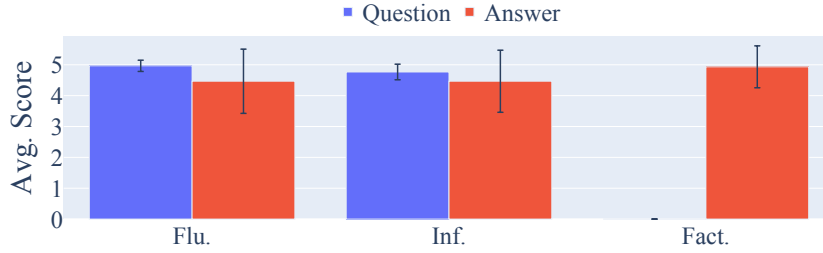


FIGURE 3.2. Human analysis on SYN-LEQA’s data quality according to the average *fluency*, *factualness*, and *informativeness* scores.

tion, 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.] and POLICYQA [Ahmad et al.], 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.

LLM Annotation. We use LLAMA-2-13B as LLM teacher $\mathcal{M}$.³ Specifically, given

³Although ACE-ATTORNEY is model-agnostic, our infrastructure is restricted to 24 GB of GPU RAM.

TABLE 3.3. Prompts engineered for synthetic data generation (above) and zero-shot LLMs (below).
 $\dagger$ denotes the final prompt chosen for the creation of SYN-LEQA.

Setting	Prompt
Synthetic Data Generation	
Task Description $\dagger$	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]
In-Context Learning	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:
In-Context Learning 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 LLMs	
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."

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 Table 3.3), 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 through 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 et al., 1995] (more details are given in subsection 3.4.4.3). Finally, to verify C3, we conduct a human analysis to gauge data quality. Specifically, we present 50 random samples to three English-proficient annotators with legal background 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 3.2 shows very high results,

Thus, we can afford to run models up to 13B in size, also simulating a real-world hardware-constrained scenario.

TABLE 3.4. 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 important 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

supported by inter-annotator agreement of 0.62 with Cohen’s κ coefficient, revealing the great generative capacity of our LLM teacher. The evaluation criteria are detailed in Table 3.4.

3.4.3.2 Selective Generative Paradigm

The quality of the generated answer is highly dependent on the retriever $\mathcal{R}$, which prevents the generator $\mathcal{G}$ 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., 2021, Moro et al., 2022a, 2023e, Lai et al., 2023, Moro et al., 2023b]. However, $\mathcal{G}$ still places complete trust in $\mathcal{R}$, without having the possibility to replace the received document.

Inspired by communication networks that require adaptability to manage dynamic scenarios [Moro and Monti, 2012b], we advocate for a $\mathcal{G}$ capable of doubting the conduct of $\mathcal{R}$. We term this ability as *Selective Generative Paradigm* (SGP) (see Figure ??), 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 **s**: ‘‘Please provide the correct document.’’ Therefore, the second most ranked document is passed to $\mathcal{G}$, which re-

Algorithm 1 Selective Generative Paradigm

Input: $\mathbf{q}$ {question}**Output:** $\mathbf{a}$ {answer}0: $\{\mathbf{d}'_1, \dots, \mathbf{d}'_n\} = \mathcal{R}(\mathbf{D} \mid \mathbf{q}), i = 1$ 0: **while** $\mathbf{a} = \mathbf{s}$ **do**0: $\mathbf{a} = \mathcal{G}(\mathbf{q}, \mathbf{d}'_i)$ 0: $i = i + 1$ 0: **end while** $= 0$

assesses the relevance until it generates a proper answer. The formal procedure is given in Algorithm 1.

Regarding zero-shot LLMs, this approach is implemented by modifying the original prompt as described in Table 3.3. Differently, SLMs undergo an altered fine-tuning process (see the dataset refinement in Figure ??). Specifically, we tamper the training set instances whose corresponding document is not correctly retrieved at the first position by BM25 OKAPI [Robertson et al., 1995] (see subsection 3.4.4.3) by replacing their target answer with $\mathbf{s}$. Then, we fine-tune SLMs on this modified dataset, called SYN-LEQA_{sgp}, whose altered training samples are 654 out of 3854.

3.4.4 Experimental Setup

Based on [Moro et al., 2018b, Domeniconi et al., 2017], in addition to the in-domain experiment on the proposed SYN-LEQA dataset, we conduct a cross-domain evaluation on the PRIVACYQA dataset [Ravichander et al.], aiming to assess the performance of the distilled SLMs trained on SYN-LEQA. Since PRIVACYQA is centered on a classification task, we make adjustments to align it with our specific generative QA objective. In fact, in this dataset, 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 $\mathbf{a}_i$. We call this testbed PRIVACYQA_{gen} and outline its statistics in Table 3.1. Since the questions did not include enough context for the retriever to behave as expected, we do not consider $\mathcal{R}$ in the pipeline.

In summary, in the *in-domain scenario*, we use SGP and therefore train SLMs on SYN-LEQA_{sgp}. Differently, in the *cross-domain scenario*, we directly feed the gold document to both SLMs and LLMs without leveraging SGP. Hence, we train SLMs on the original SYN-LEQA dataset and test on PRIVACYQA_{gen}.

Environment. All runs are tracked with Weights & Biases⁴ 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 portability, our environment is built on

⁴<https://wandb.ai>

TABLE 3.5. Hyperparameters initialization for utilized NLG metrics.

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

top of a docker container with an NVIDIA image.⁵ 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].

3.4.4.1 Metrics

To thoroughly assess model performance, we cover multiple evaluation dimensions. Table 4.11 lists the hyperparameters of the metrics.

Syntactic. 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. [2023a], we also measure an aggregated ROUGE judgment to penalize results with discrepant unigram, bigram, and longest common subsequence overlaps: $\mathcal{RG} = \text{avg}(r_1, r_2, r_L) / (1 + \sigma_r^2)$, where σ_r^2 is the F1 variance. Accordingly, we apply the same principle to BLEU, obtaining $\mathcal{B} = \text{avg}(b_1, b_2, b_3, b_4) / (1 + \sigma_b^2)$.

Semantic. Moving beyond lexical superficiality, we employ **BERTScore** (BeS) [Zhang et al., 2020c] and **BARTScore** (BaS) [Yuan et al., 2021a] for semantic coverage, two model-based metrics that exhibit strong correlations with human judgment. 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 [Lewis et al.]. 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 the hypothesis ($\mathbf{h}$). Precision ($\mathbf{r} \rightarrow \mathbf{h}$, $p(\mathbf{h}|\mathbf{r}, \theta)$) assesses how likely the answer hypothesis could be constructed based on the gold reference. F1 score ($\mathbf{h} \leftrightarrow \mathbf{r}$) takes the harmonic mean of recall and precision.

⁵nvidia/cuda:11.3.1-devel-ubuntu20.04

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	

Hyperparameter	Value
Dropout rate [†]	0.1
Learning rate [†]	$5e^{-5}$, linear scheduler
AdamW Opt. [Loshchilov and Hutter] [†]	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
Temperature	1.0
Quantization [‡]	8-bit

FIGURE 3.3. *Left*: HuggingFace checkpoints of the models used in this study. *Right*: Hyperparameters used for SLMs fine-tuning and inference, and zero-shot LLMs. [†] and [‡] are specific for SLMs and LLMs, respectively.

Efficiency. We monitor the **inference runtime** and compute **CO₂ emissions** with CodeCarbon.⁶ Finally, we condense our judgment to weigh both cost and effectiveness in a single score using **Carburacy** [Moro et al., 2023a].

3.4.4.2 Models

Figure 3.3 shows the models used in this study, specifying their checkpoints and hyperparameters for fine-tuning and inference.

SLMs. In line with previous works [Ragazzi et al., 2024a, Italiani et al., 2024], we consider **Bart** [Lewis et al.], one of the most popular models for generative tasks characterized by a denoising pretraining objective, and **Flan-T5** [Chung et al., 2022a], 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. For the fine-tuning process, we concatenate the question and the relevant document and train them to produce the corresponding answer. We train each model for 5 epochs and select the checkpoints that obtain the highest ROUGE-1 score on the validation set.

LLMs. We examine **LLaMA-2-{7B,13B}** [Touvron et al., 2023b], which is an up-graded version of LLaMA-1 [Touvron et al., 2023a] pretrained on additional 40% of data. **Mistral-7B** [Jiang et al., 2023a] comprises grouped-query attention [Ainslie et al., 2023] and sliding window attention [Beltagy et al., 2020b] 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

⁶<https://github.com/mlco2/codecarbon>

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** [Zhu et al., 2023] 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. To make inference for zero-shot LLMs possible with our hardware, we exploit PEFT.⁷ To be precise, we carry out 8-bit model quantization, enclosing the provided prompt within the recommended template for every model.

3.4.4.3 Retriever

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 [Frisoni et al., 2022a]. 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., 1995], a probabilistic model that estimates the relevancy of documents w.r.t. a query as follows:

$$\text{BM25} = \sum_{t \in q} \log \left(\frac{n - f_t + 0.5}{f_t + 0.5} \right) \cdot \text{TF}_{\text{BM25}} \quad (3.1)$$

$$\text{TF} = \frac{f_{t,\mathbf{d}} \cdot (k_1 + 1)}{f_{t,\mathbf{d}} + k_1 \cdot ((1 - b) + (b \cdot \ell_{\mathbf{d}} / \ell_{\text{avg}}))} \quad (3.2)$$

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, 2019b], which uses a siamese BERT network to create semantically meaningful sentence embeddings. This involves independently processing $\mathbf{q}$ and $\mathbf{d}_i$ through an encoder-only transformer, resulting in two unique sentence embeddings that can be compared using cosine similarity [Frisoni et al., 2020, Domeniconi et al., 2016]. To enhance computational efficiency, we conduct indexing using FAISS [Johnson et al., 2019], which tackles the problem of efficient comparison between high-dimensional vectors. Lastly, we assess the *cross-encoder* (CE) implementation [Nogueira and Cho, 2019], using an encoder-only transformer to process the concatenated $\mathbf{q}$ and $\mathbf{d}_i$, producing a score indicating the document pertinence. Looking at the MS Marco Passage Reranking [Nguyen et al.] 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.] and

⁷<https://github.com/huggingface/peft>

TABLE 3.6. Retrieval results. Bold and underline denote the best and second-best scores. $\uparrow$ =higher is better; $\downarrow$ =lower is better.

Model	R@1 $\uparrow$	R@5 $\uparrow$	R@10 $\uparrow$	Run. $\downarrow$	Emis. $\downarrow$
BM25 OKAPI	83.02	95.65	97.10	10.89	1.09
BiE DISTILROBERTA	70.81	87.78	90.89	<u>12.20</u>	<u>1.44</u>
BiE MPNET	73.50	87.16	91.72	18.02	2.15
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 3.7. Results of FLAN-T5-large on PRIVACYQA_{gen}’s test set fine-tuned on 100 instances from four versions of SYN-LEQA, each generated using different prompt templates. Bold and underline denote the best and second-best results.

Prompt	$\mathcal{RG}$	BeS	Run.
In-context learning	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
In-context learn. w/ task descr.	22.86	25.87	5061

MiniLM [Wang et al., 2020b] as cross-encoders.

Results. Table 3.6 compares the performance of different retrieval methods on the test set of SYN-LEQA. We report the Rank@ k metric, with $k \in \{1, 5, 10\}$. Technically, given an input question, we assess how frequently the corresponding document is correctly found within the top- k items retrieved. We also provide the runtime in seconds and the CO₂ emissions in milligrams. According to Reimers and Gurevych [2019b], cross-encoders exhibit the best results at the price of considerably higher computational costs because they require computing a score for every possible combination of query and document. This assertion is also validated by our case study. For example, although CE MINILM demonstrates superior performance w.r.t. BM25 OKAPI (90.06 vs. 83.02 for Rank@1), it requires more than 100x runtime (1428.88 vs. 10.89 seconds) and emissions (162.58 vs. 1.09 milligrams). Additionally, the performance gap decreases sharply as k increases (97.52 vs. 95.65 for Rank@5). For these reasons, to strike the best equilibrium between performance and computational overhead, we select BM25 OKAPI as the retriever for our in-domain experiments.

3.4.5 Results and Discussion

Prompt Engineering. As emphasized in previous studies [Khashabi et al., Wei et al., 2022, Ye et al., 2022], the configuration of the prompt template can profoundly influence the process of synthetic data generation. Therefore, we select 100 documents equally drawn from lawsuits and policies. From these documents, we generate four different versions of SYN-LEQA, with each variant differing based on the specific approach used

TABLE 3.8. Quantitative results on SYN-LEQA’s test set. Bold and underline denote the best and second-best results. * indicates statistically significant higher scores compared to the best-performing LLM (STARLING) based on a one-tailed t-test (p-value= 0.05).

Syn-LeQA													
Model	Size	Release	Syntactic					Semantic		Efficiency			R@1
			$\mathcal{B}$	R-1	R-2	R-L	$\mathcal{RG}$	BaS-F1	BeS	Carb.	Run.	Emis.	
Fine-tuned SLMs													
FLAN-T5-small	60M	2022/10	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	52.51	65.85	53.77	65.34	61.46	-1.79	66.06	80.78	249.15	9.86	89.03
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

to generate the synthetic questions and answers. These variations are produced by applying four distinct prompt engineering strategies: (see Table 3.3): (i) *Task Description*, we provide a textual description of the task and a test example; (ii) *In-Context Learning*, the model is given a few demonstrations of the task at inference time, but no weight updates are allowed; (iii) *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) *In-Context Learning with Task Description*. We then fine-tune FLAN-T5-large on these four different reduced SYN-LEQA’s versions and test it on PRIVACYQA_{gen}. With this experiment, we aim to test which prompt engineering technique generates the most effective synthetic dataset, ultimately leading to optimal cross-domain performance on PRIVACYQA for a model fine-tuned on that data. Table 3.7 indicates that *Task Description* performs the best, while *LLAMA Task Description* exhibits slightly higher efficiency, albeit with reduced effectiveness compared to the former. As expected, strategies that use in-context learning take longer to generate question-answer pairs because the model is provided with lengthier instructions as input. Moreover, this appears to harm the data generation process, resulting in extensively inferior results compared to methods that abstain from in-context learning. Consequently, all experiments are conducted with the prompt template of *Task Description*.

Effectiveness. Table 3.8 shows the in-domain results on SYN-LEQA using SGP. Across both syntactic and semantic metrics, distilled SLMs consistently outperform LLMs. The greatest improvement is registered 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, unlike LLMs, SLMs excel

TABLE 3.9. Classification results of SGP. Bold and underline denote the best and second-best results. Following standard practice [Frisoni and Moro, 2021], we evaluate model performance using precision, recall, F1-score, and accuracy metrics.

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

at identifying documents that have not been retrieved properly. The ability of the models to classify the retrieved documents as relevant or not is summarized in Table 3.9. 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 on average solidly increases the original R@1 score of 83.02, as shown in Table 3.6. The enhanced retrieval performance of SLMs also explains their superior question-answering effectiveness w.r.t. LLAMA-2-13B. Since this LLM was used to produce SYN-LEQA, superior results would be expected when evaluated on the same questions it generated. However, its performance is hindered by a markedly decreased R@1 score, which ultimately results in less informed answers. Furthermore, Table 3.10 shows the cross-domain results on PRIVACYQA_{gen}, where we test SLMs trained on SYN-LEQA. Again, SLMs overall outperform the majority of LLMs, albeit with a less pronounced improvement. First, this occurs because we transfer the question-answering capabilities of the models, acquired on SYN-LEQA, to a completely different dataset. Second, this setting does not include the retrieval module. In scenarios where the retrieval component is not needed, SLMs cannot benefit from SGP, hence the improvement w.r.t. LLMs shrinks. Moreover, we highlight that even when the best-performing SLM underperforms compared to the top LLM, the difference is not statistically significant except for R-1. Figure 3.4 visually summarizes the results.

Efficiency. In addition to the better effectiveness of distilled SLMs, from Table 3.8 and Table 3.10, we can also appreciate the radical reduction in runtime and carbon emissions

TABLE 3.10. Quantitative results on the test set of PRIVACYQA_{gen}. Bold and underline denote the best and second-best results. * indicates statistically significant higher performance compared to the best-performing LLM (STARLING) based on a one-tailed t-test (p-value= 0.05)

PrivacyQA _{gen}												
Model	Size	Release	Syntactic					Semantic		Efficiency		
			β	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

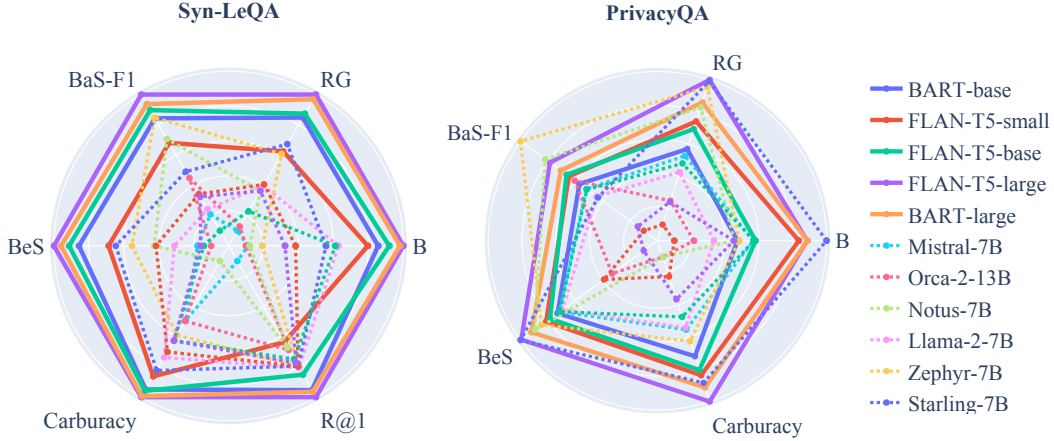


FIGURE 3.4. Graphical overview of model effectiveness, where dotted lines represent LLMs and solid lines SLMs.

at inference time. For example, the best-performing LLM on PRIVACYQA_{gen}, such as STARLING-7B, is characterized by $\approx 1200\%$ increment in CO₂ emissions compared to the best SLM, such as FLAN-T5-large. Figure 3.5 illustrates the relationship between efficiency and effectiveness, revealing the high performance of SLMs with low costs.

Few-shot Setting. Limitations due to low-resource regimes, such as commodity hardware infrastructure, can affect model performance [Parida and Motlíček, Moro and Ragazzi, 2022a, 2023, Huh and Ko, 2023, Moro et al., 2023f]. Regardless, it is plau-

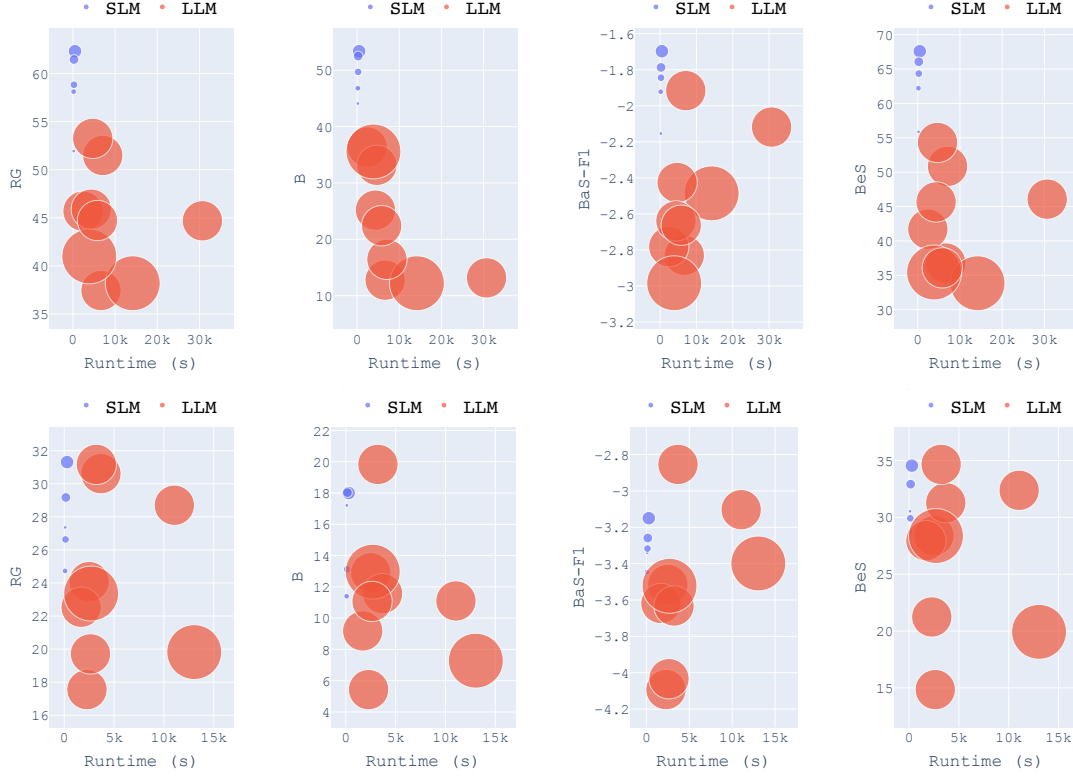


FIGURE 3.5. Performance–efficiency relation. Top: SYN-LEQA; bottom: PRIVACYQA_{gen}. The size of bubbles refers to the number of model parameters.

sible that a real-world organization lacks adequate 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 3.6 summarizes the results of FLAN-T5-large fine-tuned on an increasing number of SYN-LEQA samples, drawn equally from lawsuits and policies. Notably, in SYN-LEQA, the SLM surpasses the best-performing LLM, such as STARLING-7B, with only 20 labeled samples. In PRIVACYQA_{gen}, we notice the same behavior in 2 out of 4 metrics, such as BLEU and BARTScore.

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 testing as upper bound. Figure 3.7 demonstrates the adaptability of SLMs, showcasing consistent ROUGE performance when transferred between different domains. Additionally, we perform an experiment to assess the extent to which a model can leverage knowledge acquired during its initial training on SYN-LEQA and transfer it to improve performance on

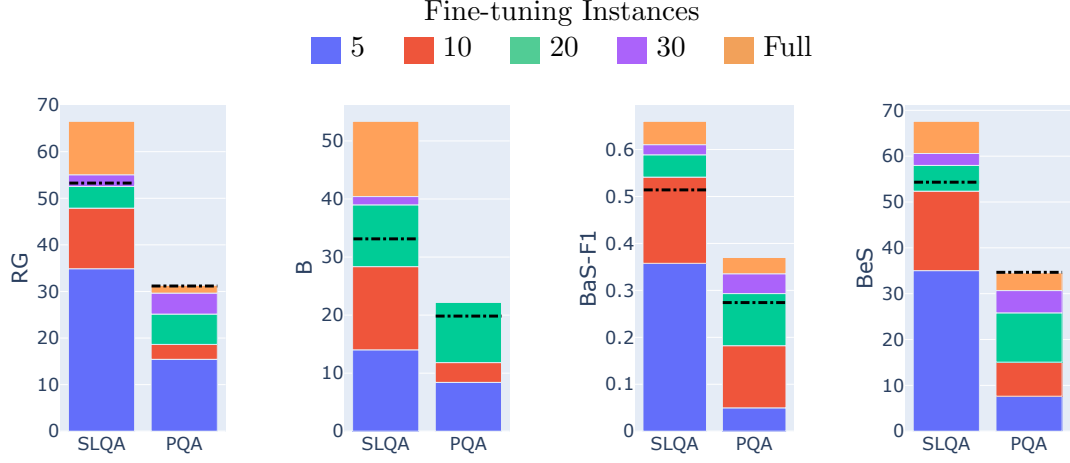


FIGURE 3.6. Few-shot setting results for SYN-LEQA (SLQA) and PRIVACYQA_{gen} (PQA). The dashed black bands mark the top-performing LLM (STARLING-7B).

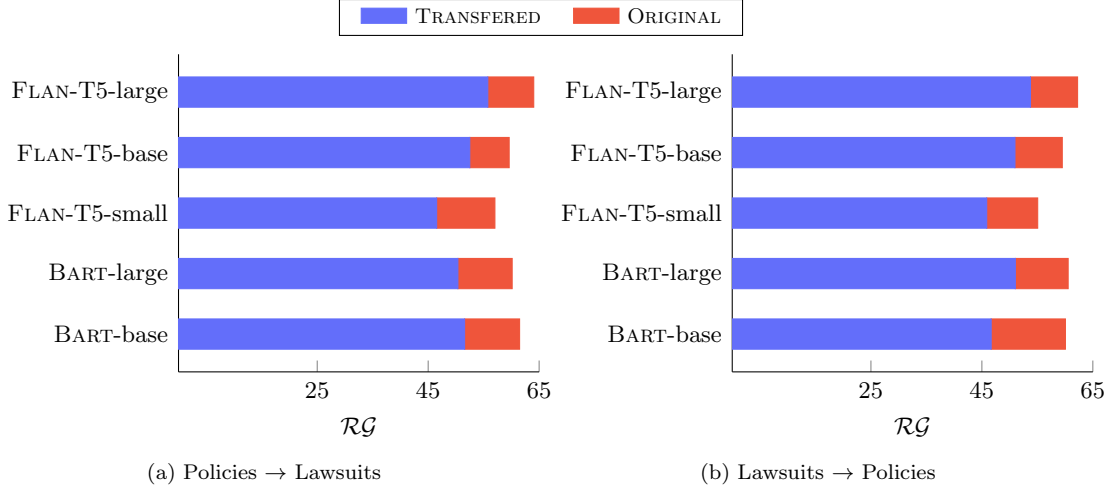


FIGURE 3.7. Transferability results on SYN-LEQA. TRANSFERED denotes a model trained on one subset and tested on the other. ORIGINAL denotes a model trained and tested on the same subset.

PRIVACYQA_{gen}. Specifically, we first train the best-performing model, FLAN-T5-large, on our synthetic dataset, and then fine-tune it on PRIVACYQA_{gen}. For this purpose, we divide the original PRIVACYQA_{gen} dataset, which originally contains only a test set, into three subsets: 344 instances for training, 43 for validation, and 44 for testing. Table 3.11 demonstrates that the model fine-tuned solely on PRIVACYQA_{gen} performs worse than when it is initially fine-tuned on SYN-LEQA, underscoring the usefulness of our synthetic dataset.

TABLE 3.11. FLAN-T5-large performance on the test set of PRIVACYQA_{gen}.

Training Set	$\mathcal{B}$	R-1	R-2	R-L	$\mathcal{RG}$	BaS-F1	BeS
PRIVACYQA _{gen}	80.68	85.36	82.33	85.52	84.39	-0.972	85.53
SYN-LEQA + PRIVACYQA _{gen}	80.49	90.10	88.35	90.58	89.67	-0.717	90.43

3.4.6 Conclusion

In this thesis, we introduce ACE-ATTORNEY, an approach designed to produce LQA-specific datasets and supervised SLMs through LLM KD. Precisely, given an input textual prompt, a 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 showcase the ability of student models to surpass the latest cutting-edge LLMs. Importantly, distilled models operate with amply fewer parameters and demand considerably shorter inference runtime and CO₂ emissions. Moreover, simulating a real-world retrieval-based scenario, we show that the Selective Generative Paradigm enables the retriever and the generator to better select the correct document to answer a given question. Finally, we reveal that our specialized SLMs achieve competitive performance even when we have access to a limited number of synthetic instances, allowing manual oversight at a reasonable cost across the entire generated dataset.

Ethics Statement and Limitations

Our research holds noteworthy promise for advancing the development of new LQA datasets and models. However, given the widely recognized issue of hallucinations in LLMs, experts must examine the authenticity and reliability of the generated data. In fact, our aim is not to demonstrate that LLMs could replace legal professionals. Instead, we introduce ACE-ATTORNEY as a research instrument, highlighting its complementary function rather than positioning it as a substitute for detailed and contextually specific efforts.

Despite the large amount of up-to-date LLMs used in our experiments, we focus on models capable of working within the constraints of a single 24 GB of GPU RAM. Consequently, this excluded larger and potentially more performant solutions from our study, which could affect the synthetic data generation process. However, human examination suggested high quality despite the use of a 13B LLM. Furthermore, the cross-domain scope of our study regarding the evaluation of our fine-tuned SLMs on real human-annotated datasets is constrained, as we considered only a single dataset in our analysis. However, this decision is due to PRIVACYQA_{gen} being the only dataset that meets the necessary criteria to act as a benchmark for generative LQA. Finally, due to the limited availability of multilingual legal datasets [Niklaus et al.] and solutions [Moro et al., 2024a], future directions should prioritize advances in low-resource languages with a focus on interpretable responses [Moro et al., 2024b].

Chapter 4

Optimization through Minimal Architectural Modifications in Large Language Models

4.1 Introduction

The preceding chapters have examined two principal optimization paradigms in large language models (LLMs): knowledge distillation, which transfers knowledge from a larger teacher model into a smaller or more efficient student, and token pruning, which dynamically reduces redundant computation during inference. Both techniques target efficiency—reducing computational or memory demands while preserving performance.

In contrast, the present chapter explores a complementary perspective on optimization: techniques that enhance the performance of baseline architectures through minimal structural modifications. These methods do not primarily aim to compress or accelerate models, but rather to optimize their representational and inductive capacity by altering the underlying architecture in subtle, computationally economical ways. The unifying principle is that small architectural interventions, such as introducing a gating mechanism, modifying normalization strategy, or reparameterizing attention, can yield measurable improvements in generalization, robustness, and accuracy. This direction of research aligns with a broader trend in neural architecture optimization: understanding how modest, well-placed changes in model topology can amplify learning dynamics. As such, this chapter positions these modifications as legitimate forms of optimization, since they optimize the model’s functional capacity rather than its resource footprint.

4.2

Cogito Ergo *Summ*: Abstractive Summarization of Biomedical Papers via Semantic Parsing Graphs and Consistency Rewards

Originally published as: 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 *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 12781–12789, 2023b

Abstract

The automatic synthesis of biomedical publications catalyzes a profound research interest elicited by literature congestion. Current sequence-to-sequence models mainly rely on the lexical surface and seldom consider the deep semantic interconnections between the entities mentioned in the source document. Such superficiality translates into fabricated, poorly informative, redundant, and near-extractive summaries that severely restrict their real-world application in biomedicine, where the specialized jargon and the convoluted facts further emphasize task complexity. To fill this gap, we argue that the summarizer should acquire semantic interpretation over input, exploiting structured and unambiguous representations to capture and conserve the most relevant parts of the text content. This thesis presents COGITOERGOSUMM, the first framework for biomedical abstractive summarization equipping large pre-trained language models with rich semantic graphs. Precisely, we infuse graphs from two complementary semantic parsing techniques with different goals and granularities—Event Extraction and Abstract Meaning Representation, also designing a reward signal to maximize information content preservation through reinforcement learning. Extensive quantitative and qualitative evaluations on the CDSR dataset show that our solution achieves competitive performance according to multiple metrics, despite using $2.5\times$ fewer parameters. Results and ablation studies indicate that our joint text-graph model generates more enlightening, readable, and consistent summaries.

4.2.1 Introduction

Given the sheer number of biomedical publications, clinicians, patients, and researchers need advanced tools to skim the literature efficiently and grasp salient contents. Hence, automatically organizing everyday scientific discoveries or insights into natural, concise, and informative syntheses is essential to promote knowledge acquisition [Moradi and Ghadiri, 2019]. To this end, abstractive document summarization demands rephrasing and condensing long and often labyrinthine portions of text in a creative way, discarding redundant and unnecessary attributes. Compared with the open domain, performing this task in biomedicine raises substantial challenges and constraints [Karamanis, 2007].

Indeed, (i) medical jargon and professional language are truly hard to interpret; (ii) scientific documents convey precise domain information allowing for a narrow interpretation margin and non-tolerating factual mistakes; (iii) clauses are often interdependent and express complex interactions; (iv) knowledge rapidly evolves over time.

Despite unprecedented progress made possible by pre-trained transformer-based language models [Lin and Ng, 2019], current summarizers still face issues in terms of succinctness, non-repetitiveness, fluency, informativeness, and faithfulness [Maynez et al., 2020]. Most prior studies only depend on superficial text organization and ignore the deeper underlying semantic content [Bender et al., 2021], lacking structured representations to encapsulate the convoluted long-range associations between the mentioned entities (e.g., proteins, diseases, drugs). By contra, we argue that these connections are vital to document understanding and beneficial to knowledge selection.

Biomedical documents are usually composed of a series of events and factual evidence; understanding how to leverage such information in generative models is crucial. Notably, semantic parsing graphs normalize many lexical and syntactic variations by providing formal meaning representations capable of decoupling concept units (*what to say*) from language competencies (*how to say it*). Since these representations can be defined with a panoply of formalisms having different objectives and properties, we underline the importance of bridging the complementary strengths of two influential semantic parsing tasks: closed-domain Event Extraction (EE) and Abstract Meaning Representation (AMR). EE is task-driven and aims to derive n-ary and potentially nested interactions between participants having a specific semantic role, where event schemas (i.e., target event, entity, and role types) are pre-established conforming to a reference ontology; its history is very intertwined with health informatics [Frisoni et al., 2021]. AMR is linguistically-grounded and is conceived to graphically capture the general meaning of any sentence as high-level semantic relations between abstract concepts [Banarescu et al., 2013]. Fig. 4.1 depicts and compares their expressive power.

A growing body of research in natural language generation (NLG) calls attention to incorporating explicit semantic structures into the summarization process [Yu et al., 2020], thereby unlocking a higher level of abstraction than bags of sentences and more accurate emulation of human interpretation, rewriting, and paraphrasing. However, existing graph-augmented approaches have at least one of the following weaknesses: (i) they have not been designed for or evaluated in the biomedical domain; (ii) they employ graph-LSTMs architectures that struggle to compete with transformers; (iii) they are based on open-domain triplet-based extractions that are notoriously not adequate to represent the complete biological meaning of a document; (iv) they do not include a module to ensure document-summary consistency.

We present COGITOERGOSUMM¹, the first semantics-aware transformer-based model for single-document abstractive summarization in the biomedical domain. Concretely, we condense source documents into sets of unambiguous EE and AMR graphs, using

¹Inspired by the first principle of René Descartes’s philosophy, we coin the name *Cogito Ergo Summ* to emphasize the researched neural network capacity of “thinking” about the inner semantics of the text—via joint text-graph reasoning—before summarizing.

multi-relational graph neural networks (GNNs) to yield their dense embeddings without imposing linearization. From this foundation, we explore a fine-tuning recipe for merging predicted symbolic representations into pre-trained encoder-decoder language models. Specifically, we integrate semantic parsing graphs via EE- and AMR-specific attention mechanisms in the decoder, thus aiding key content selection and semantic understanding. We optimize the network via reinforcement learning (RL), devising a consistency-guided reward signal based on a soft alignment between the sets of meaning representations extractable from documents and generated summaries.

Automatic and human evaluations are carried out on CDSR [Guo et al., 2021b], a popular dataset for generating lay language summaries of biomedical reviews. Our model brings substantial improvements in compliance with multiple quality criteria, achieving near state-of-the-art (SOTA) performance while using 2.5x fewer parameters. Empirical results corroborate the value of semantic graphs in helping the model to preserve the essential global context and keep the factual connections between the most relevant entities.

4.2.2 Related Work

Abstractive Summarization Sequence-to-sequence architectures founded on self-supervised pre-trained transformers have been responsible for a profound impetus in implicitly learning abstractive summarization procedures [Zhang et al., 2020b], comprising multi-document [Moro et al., 2022b] and low-resource settings [Moro and Ragazzi, 2022b]. Nevertheless, modern solutions are highly prone to hallucinating content [Cao et al., 2018, Maynez et al., 2020] or falling back on extraction [See et al., 2017]. Gaining an understanding of semantics and context is becoming a prerogative, but a model trained purely on form cannot learn meaning [Bender and Koller, 2020]—even with more data and huge architectural dimensions.

Graph-enhanced Summarization Graph structures have long been studied for implementing summarization subtasks (i.e., information extraction, content selection, surface realization), registering distinct benefits depending on their composition. Early techniques for extractive summarization build intra- [Mihalcea and Tarau, 2004] and inter-document [Wan, 2008] cosine similarity connectivity networks to identify salient sentences. Late hybrid neural systems mainly stand on the shoulders of run-of-the-mill GNNs [Wu et al., 2021b], exploiting graph-based attention [Tan et al., 2017] and heterogeneous word-/sentence-level nodes [Wang et al., 2020a]. As for abstractive summarization, the community has attempted a medley of graphical document representations, from dependency [Wu et al., 2021c], sentiment [Moro et al., 2018a], and coreference links [Balachandran et al., 2021] to discourse relations [Li et al., 2020, Chen and Yang, 2021b] and citation networks [An et al., 2021a]. To better contemplate entity interactions, compositions of $\langle \textit{subject}, \textit{predicate}, \textit{object} \rangle$ triplets from the OpenIE framework [Angeli et al., 2015] have turned into a cornerstone [Fan et al., 2019, Huang et al., 2020, Ji and Zhao, 2021, Zhu et al., 2021]. On the flip side, open-domain binary relations are

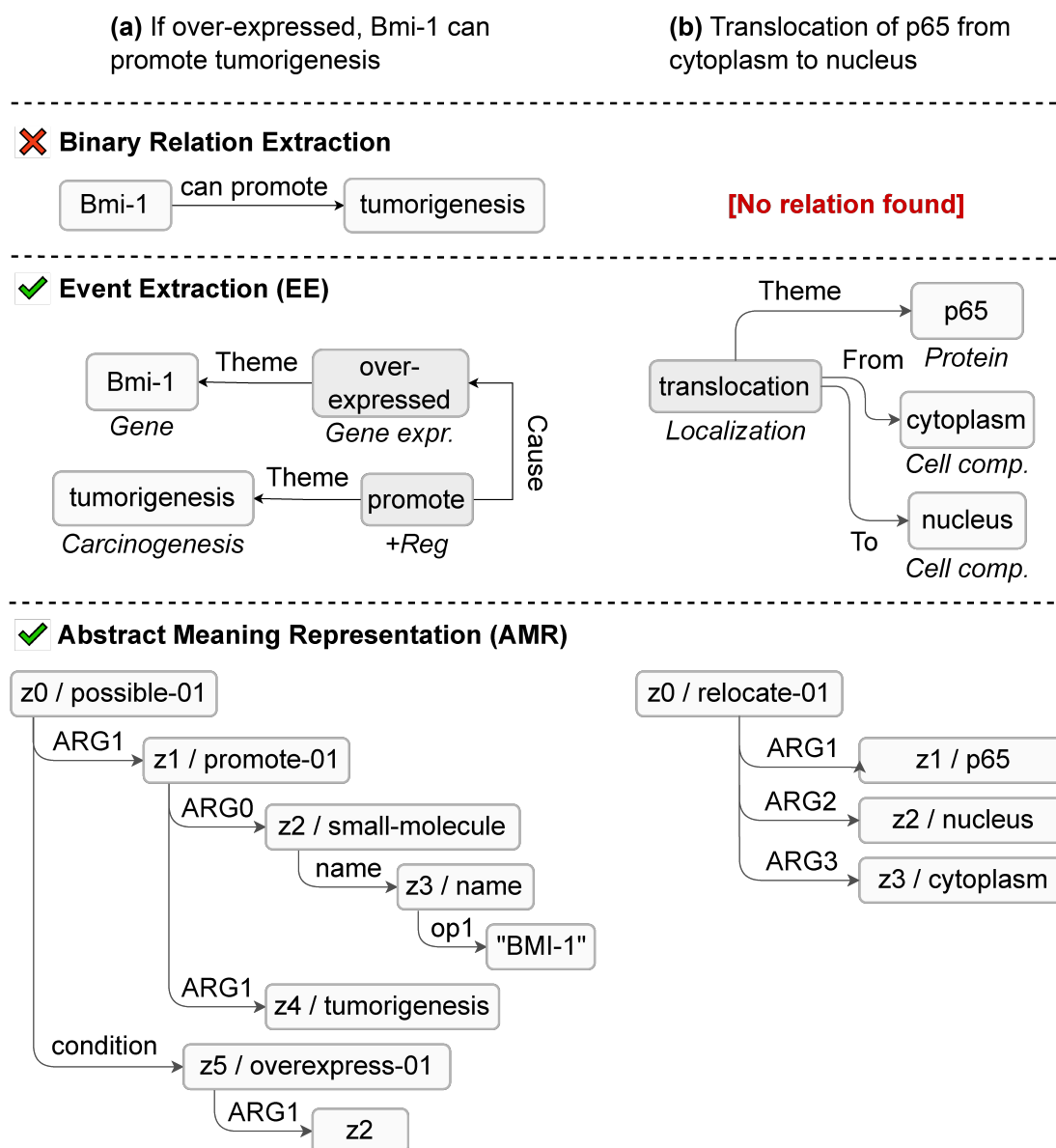


FIGURE 4.1. Expressiveness comparison between binary relation extraction (OpenIE 5.1), EE, and AMR on two example biomedical sentences. *Italic* denotes node types.

inadequate for biomedicine, risking deriving incorrect or incomplete facts that are difficult to compare and merge with post-processing [Frisoni et al., 2021]. Most pertinent to our work are summarizers enhanced by AMR semantic analysis [Dohare et al., 2018, Hardy and Vlachos, 2018, Lee et al., 2021a], underexplored in bioinformatics despite showcasing superior generation controllability in general domain. Frisoni. et al. [2022] investigate EE-augmented summarization observing that performances are held back by

the reduced number of event graphs. Furthermore, almost all solutions nowadays reckon on graph-LSTM architectures—that hardly compete with transformers—or graph-to-text verbalizers—that ignore source text contribution. To our knowledge, we are the first to combine text, EE, and AMR for transformer-based abstractive summarization, solving their mutual limitations. Another trend is context augmentation through external knowledge graphs, such as UMLS for medicine [Giglioli et al., 2018]. However, unlike flexible meaning representations, these resources are known to have limited and static coverage of real-world entities.

Reinforcement Learning for Abstractive Summarization In a traditional encoder-decoder architecture, the network is trained to minimize the maximum-likelihood loss for next-token prediction but is evaluated on the not necessarily equivalent optimization of desired automatic metrics. Moreover, the decoder knows the ground-truth sequence during training but does not have such supervision when testing, leading to an exposure bias [Ranzato et al., 2016]. RL methods have recently been prompted to mitigate these discrepancies by directly solving metric optimization problems that are not differentiable, not requiring gold summaries, and allowing global decisions rather than local (token-level) ones at each timestep. Kryscinski et al. [2018], Paulus et al. [2018], Sharma et al. [2019] utilize ROUGE scores to encourage novelty and relevance, ignoring entity interactions. Scialom et al. [2019], Huang et al. [2020] propose fill-in-the-blank and triplet-reconstruction question answering rewards, needing ad-hoc models for generating artificial questions. In contrast, we reward predicted summaries according to their key content consistency with respect to the original document, measuring the alignment between their semantic graphs.

4.2.3 CogitoErgoSumm Framework

4.2.3.1 Problem Statement

Given a dataset $C=(d_1, d_2, \dots, d_k)$, each document d_i consists of a sequence of n tokens $d=(x_1, x_2, \dots, x_n)$. The semantics of d_i is condensed in document-level event and AMR graphs (G_e and G_a , respectively). Formally, the goal is to generate the target summary $y=(y_1, y_2, \dots, y_m)$, $m \leq n$, of each instance by modeling the conditional distribution $p(y_1, y_2, \dots, y_m | x_1, x_2, \dots, x_n, G_e, G_a)$, conceptualizable as a neuro-symbolic task. We keep G_e and G_a separate to let the model realize their unique characteristics. Fig. 4.2 illustrates a concrete example.

4.2.3.2 Graphs Construction

We first obtain sentence-level EE and AMR outputs, i.e., multi-relational directed acyclic graph structures symbolizing the core concepts of each sentence from two viewpoints. Then, we address graph fusion to build G_e and G_a , ensuring to end up with single wisely-interconnected document graphs rather than two sets of small disjoint networks.

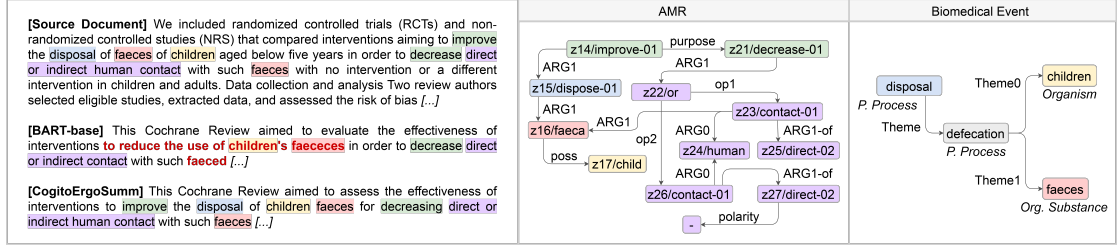


FIGURE 4.2. Qualitative example of induced semantic parsing graphs and their assistance to high-quality summarization. Red text indicates hallucinated facts. Token background highlights node alignments.

Event Graphs In consonance with BioNLP-ST competitions, events are forged from a trigger (a span that testifies their occurrence, e.g., “interacts”, “regulates”), a type (e.g., “binding”, “regularization”), and a set of arguments—classed entities or events themselves—playing a certain role (e.g., “theme”, “cause”). The specificity of the interactions sought expectedly makes EE domain-specific. We adopt DeepEventMine [Trieu et al., 2020], an end-to-end framework holding the SOTA on seven biomedical benchmarks. We convert output standoff .a* annotation files to heterogeneous event graphs following Frisoni et al. [2022c]. A node indicates a trigger or an entity, while an edge acts for an entity-trigger or trigger-trigger relation, with the second applying for nested events. Unlike AMRs, event graphs are not available for all sentences but only for the ones expressing desired interactions for which the model has been trained.

AMR Graphs AMR aims to produce a language-neutral representation of meaning, abstracting away from English and providing a layer of abstraction from words to concepts (objects, attributes, etc.) in a rooted graph. It covers ≈ 100 widespread PropBank semantic roles; as EE, annotations include entity/role identification and typing. We use a SOTA text-to-text AMR parser [Bevilacqua et al., 2021] with an 83.0 Smatch score on the AMR 3.0 (LDC2020T02) sembank. Since many EE outputs may be empty and damage model robustness [Frisoni. et al., 2022], AMRs supplement not always exploitable event graphs.

Graphs Merging and Rewiring Inside EE and AMR formalisms, an entity, trigger, or concept is canonicalized and represented by a single graph fragment, regardless of how many times it recurs in the sentence (semantic integrity). If a node fulfills multiple roles, AMRs cover within-sentence coreference edge types. On top of event and edited AMR representations, we separately operate graph rewiring to reflect the document structure and enhance the information flow. Mechanically, we introduce artificial *sentence nodes*, each connected to all the event/AMR vertices that originate from that mention. Sentence nodes are linked to each other following their positional order and collected by a *master node*. The resulting G_e and G_a graphs formulate intra- and cross-sentence information, allowing for document traversal by narrative order, concept association, or proximity.

4.2.3.3 Model

Motivated by the limitations of graph-empowered LSTMs [Frisoni. et al., 2022], COGITOERGOSUMM extends a pre-trained BART-base architecture [Lewis et al., 2020b] with the nimble ability to attend to semantic parsing graphs during decoding and preserve the most relevant information via RL. Fig. 4.5 sketches the overall architecture, built upon four modules, namely text encoding, graph encoding, semantics-driven multi-view decoding, and consistency rewarding.

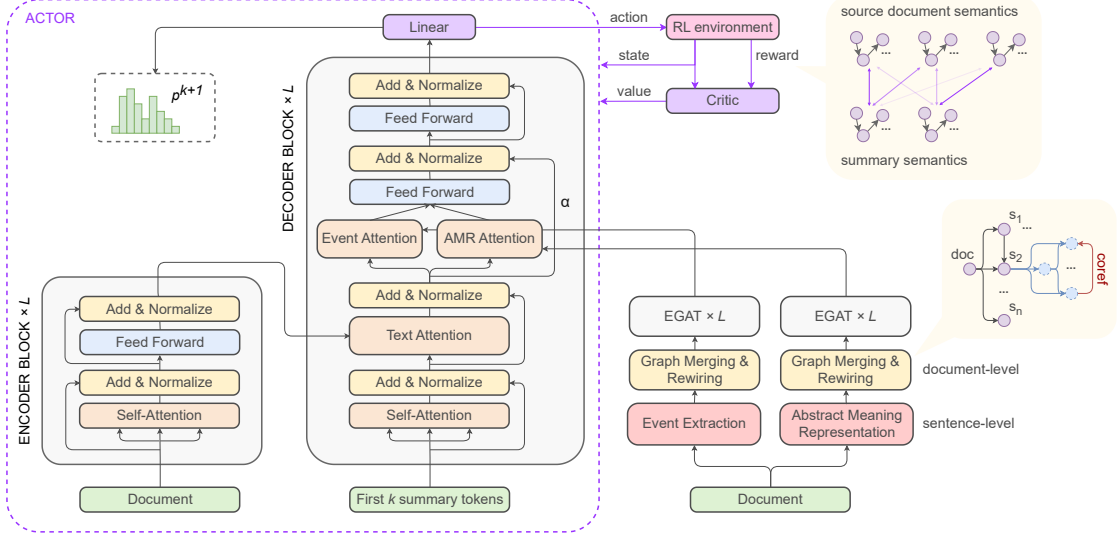


FIGURE 4.3. Illustration of the COGITOERGOSUMM architecture. We dig out semantic parsing graphs from the input text by means of event extraction and AMR. Each source document is fed to a transformer encoder; event and AMR graph embeddings are learned by edge-aware graph attention networks. A multi-granularity decoder then generates summaries based on all information levels: text, events, and abstract meaning representations. Document-summary semantic graph alignments are used to optimize model consistency via reinforcement learning.

Text Encoder We feed the input document to a text bidirectional encoder $E_t(\cdot)$ through the learnable BART encoding channel. The l tokens of a record d_i are converted in their contextual hidden representations:

$$\{h_{t_{i,0}}, \dots, h_{t_{i,l}}\} = E_t(\{x_{i,0}, \dots, x_{i,l}\}). \quad (4.1)$$

Graph Encoders *Node Initialization.* Since all nodes are accompanied by text, we initialize their features with embeddings from a pre-trained BART language model $E_n(\cdot)$. Pointedly, given a trigger/entity/concept node i with a tokenized text attribute $\text{tokens}(i)$, we take the average of its per-token embeddings contextualized on the long input document, i.e., $E_n([x_1, x_2, \dots, x_n \parallel \text{tokens}(i)])$, where $[\cdot \parallel \cdot]$ denotes the concatenation operator. To reduce structured prediction noise, we set the maximum node length to 5 tokens. For event graphs, we also prepend the entity and trigger types learned by Deep-EventMine. Finally, for sentence and master nodes, we average the token embeddings of the sentence span and the entire document, respectively.

Edge-Aware Graph Attention Network. Through two graph encoders $E_{G_e}(\cdot)$ and $E_{G_a}(\cdot)$, we take the multi-view semantics modeled by G_e and G_a to learn supervised node embeddings and tap implicit relations. Our GNN modules lean on the graph attention groundwork (GAT) [Velickovic et al., 2018], which induces node representations by way of L layers of message passing (shared parameters) and multi-head attention neighborhood aggregation. Message passing in graphs made of cooperating nodes Lodi et al. [2010b], Cerroni et al. [2013] is actually an established work mode borrowed from communication networks and distributed algorithms. As G_e and G_a are multi-relational graphs, we extend GAT to consider the edge type e connecting two nodes. In the ℓ -th layer, we update the representation $\vec{h}_i^{(\ell)} \in \mathbb{R}^D$ of each node i by:

$$t_{ij}^k = \sigma \left(\mathbf{a}^{\mathbf{k}}^T [\mathbf{W}^{\mathbf{k}} \vec{h}_i \parallel \mathbf{W}^{\mathbf{k}} \vec{h}_j \parallel \mathbf{W}_{\mathbf{r}}^{\mathbf{k}} \vec{e}_{ij}] \right), \quad (4.2)$$

$$\alpha_{ij}^k = \frac{\exp(t_{ij}^k)}{\sum_{z \in \mathcal{N}(i)} \exp(t_{iz}^k)}, \quad (4.3)$$

$$\vec{h}_i^{(\ell+1)} = \left\| \sum_{k=1}^K \sigma \left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^k \mathbf{W}^{\mathbf{k}} \vec{h}_j^{(\ell)} \right) \right\|, \quad (4.4)$$

where $\|_{k=1}^K$ denotes the concatenation of K attention heads, α_{ij}^k is the normalized attention weight computed by the k -th attention head, $\mathcal{N}(i)$ is the set containing the first-order neighbors of i (including i), $\vec{e}_{ij}$ is the one-hot embedding of the relation type² connecting nodes i and j , $\mathbf{W}^{\mathbf{k}}$, $\mathbf{W}_{\mathbf{r}}^{\mathbf{k}}$, and $\mathbf{a}^{\mathbf{k}}$ are trainable parameters, σ is a LeakyReLU nonlinearity.

Decoder Inspired by Chen and Yang [2021b], we aggregate different levels of encoded representations via a multi-granularity decoder $D(\cdot)$, which predicts the l -th token as:

$$\hat{y} = D(y_{<l}, E_t(d), E_{G_e}(G_e), E_{G_a}(G_a)), \quad (4.5)$$

$$P(y_l | y_{<l}, d, G_e, G_a) = \text{softmax}(\mathbf{W}_{\mathbf{p}} \hat{y}), \quad (4.6)$$

where $\mathbf{W}_{\mathbf{p}}$ stands for a trainable linear projection. We supplement the BART transformer decoder with two extra cross-attentions (Event Attention and AMR Attention), conducted over the node representations learned by E_{G_e} and E_{G_a} in parallel. We incorporate them into each layer after the original text cross-attention. The token-, event-, and -AMR attended vectors (a^t , a^e , a^a) are combined into a semantics-aware representation a^s through a feed-forward network. To accelerate the training of the new modules and alleviate the negative impact of randomly initialized graph encoders and cross-attentions at early stages, we apply ReZero [Bachlechner et al., 2021] to the residual connection after attending to semantic graphs in each decoder layer: $a^s = a^t + \alpha a^s$, where α is a learnable parameter modulating updates from cross-attention over semantic graphs.

²Given the non-talking nature of the AMR labels, we make use of categorical edge features, agglomerating numbered arguments (e.g., :op*, :quant*), except for the core ARG* ones.

Training Objectives and Consistency Reward During training, we seek to maximize the estimated probability of the actual summary. We adopt the common negative log-likelihood loss function using the teacher-forcing strategy:

$$\mathcal{L}_{nll} = - \sum \log P_{\theta}(y_l | d, G_e, G_a), \quad (4.7)$$

with θ denoting the set of model parameters. In addition to being syntactically correct, we would like the generated summaries to factually preserve as much pivotal information as possible from the original document. Drawing inspiration from the success of PICO [DeYoung et al., 2021], we believe that structured semantic representations are suitable not only for improved text generation but also for biomedical consistency evaluation. Standing on this intuition, we design a lightweight reward function ψ to maximize the non-differentiable degree of document-summary meaning overlap, which is made possible with second-stage RL training. We refer to Smatch (F-score) [Cai and Knight, 2013], metric computing the matching triples between two AMR graphs, benefiting from a higher correlation with human factuality judgments than summary-target ROUGE [Ribeiro et al., 2022]. Document-level consistency rewards are obtained via an average pooling; each AMR graph of a predicted summary sentence is compared with all the AMR graphs of the source sentences (one-to-many soft alignments):

$$\phi = \frac{\sum_{y_i \in \text{AMRs}(y)} \left(\frac{\sum_{d_i \in \text{AMRs}(d)} \text{smatch}(y_i, d_i)}{|\text{AMRs}(d)|} \right)}{|\text{AMRs}(y)|}, \quad (4.8)$$

where $\text{AMRs}(\cdot)$ stands for the sentence-level AMR graphs.

Given a starting policy π corresponding to the model trained following Eq. 4.19, we see autoregressively predicted tokens as actions and employ Proximal Policy Optimization [Schulman et al., 2017] to maximize:

$$r_t(\theta) = \frac{\pi_{\theta}(a_t | s_t)}{\pi_{\theta_{old}}(a_t | s_t)}, \quad (4.9)$$

$$\mathcal{L}_{ppo} = \hat{\mathbb{E}}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon))\hat{A}_t], \quad (4.10)$$

where $r_t(\theta)$ denotes the probability ratio between the action under the policy at iteration t (i.e., the token a_t generated conditionally to the previous ones s_t) and the action under the previous policy. $\hat{\mathbb{E}}[\dots]$ indicates the empirical average over a finite batch of samples; the clip function, combined with the ϵ hyperparameter, ensures that the policy does not change too much among iterations; $\hat{A}$ is the advantage function, an estimation of the relative improvement obtained from the selected action in the current state. To compute advantages, we use the Generalized Advantage Estimation formula Schulman et al. [2016]:

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \dots + (\gamma\lambda)^{T-t+1}\delta_{T-1}, \quad (4.11)$$

where $\delta_t = \psi_t + \gamma V(s_{t+1}) - V(s_t)$; λ and γ are hyperparameters and T is the length of the generated text. The value function V is modeled using a second neural network

trained in parallel with the main model (Actor-Critic paradigm). As proposed in Ziegler et al. [2019], we add a penalty term to the reward function that prevents the new policy from deviating too much from the already pre-trained language model by reducing in expectation their KL -divergence:

$$KL = \log \frac{\pi_{\theta}(a_t|s_t)}{\pi_{base}(a_t|s_t)}, \quad (4.12)$$

$$\hat{\psi}(a_t|s_t) = \psi(a_t|s_t) - \beta KL, \quad (4.13)$$

where π_{base} is the starting policy, and β is an adaptive coefficient (initially set via a hyperparameter) that dynamically changes during training to target a specific value of the KL -divergence (termed KL_t , see Ziegler et al. [2019]). The final loss function is:

$$\mathcal{L}_{rl}(\theta) = \mathbb{E}[\mathcal{L}_{ppo}(\theta) - c_1 \mathcal{L}_{vf}(\theta) + c_2 S[\pi_{\theta}](s_t)], \quad (4.14)$$

where $\mathcal{L}_{vf}$ is the mean square error of the value function in charge of updating the critic network; $S[\pi_{\theta}](s_t)$ is an entropy term used to ensure enough exploration during training; c_1 and c_2 are hyperparameters used to tune the importance of each component in the loss.

4.2.4 Experimental Setup

4.2.4.1 Dataset

We train and evaluate our model on the CDSR dataset [Guo et al., 2021b], a publicly available corpus acquired by the widely-used Cochrane Database of Systematic Reviews³. CDSR is intended for health literacy, assessing the automatic generation of lay language summaries from biomedical scientific reviews. As far as we know, it is the only biomedical summarization benchmark with manageable sizes and already-known graph-augmented baselines. CDSR comprises 6,677 high-quality pairs, where the source is a long abstract in professional language, and the target is a plain general-public version written by review authors or Cochrane staff. Besides creating accurate and factual summaries, this task also requires a joint style transition, imposing obstacles like terminology explanation and sentence structure simplification that outline a perfect testbed for COGITOERGOsumm. Details are presented in Table 4.1.

4.2.4.2 Implementation Details

We extend the HuggingFace implementation of BART-base⁴ and initialize weights from a model pre-trained on PubMed⁵, leaving 42 as the default training seed. We truncate input documents and set the maximum output length to 1024. We utilize the Deep-EventMine model pre-trained on MLEE⁶—the biomedical benchmark most aligned to

³<https://www.cochranelibrary.com/cdsr/reviews>

⁴https://huggingface.co/docs/transformers/model_doc/bart

⁵<https://huggingface.co/gayanin/bart-mlm-pubmed>

⁶<http://nactem.ac.uk/MLEE> (from molecules to organisms)

CDSR [Frisoni. et al., 2022]. On average, extracting all the sentence-level AMR and event graphs from a source document takes 3 and 1.2 seconds, respectively. We implement GNNs with PyTorch Geometric [Fey and Lenssen, 2019]. During RL, we freeze the encoder parameters and train the model decoder only; the critic network is a multi-layer perceptron with one hidden layer (hidden size 256). Hyperparameters are listed in Appendix. Each experiment is performed on a workstation having one Nvidia GeForce RTX3090 GPU with 24GB of dedicated memory, 64GB of RAM, and an Intel® Core™ i9-10900X1080 CPU @ 3.70GHz. Training our best model requires 20GB VRAM and 32 hours (19 for RL); 3.18 kg CO2e carbon footprint, 9.83 kWh energy needed, 25.77 Υ_m average carburacy [Moro et al., 2023c].

4.2.4.3 Baselines

We head-to-head compare COGITOERGOSUMM to representative extractive and abstractive summarization models.

- *Oracle*. It creates an extractive summary by selecting the sentences in the document having the highest ROUGE-2 score with the target (syntactic match upper bound).
- *BERT* [Liu and Lapata, 2019b]. Inter-sentence encoder with classification head, supervised by Oracle extractive.
- *Pointer generator* [See et al., 2017]. Seq2seq model trained both to copy words from the source and generate new ones from a fixed vocabulary.
- *BART* [Lewis et al., 2020b]. We take into account models pre-trained on PubMed.
- *EASumm* [Frisoni. et al., 2022]. An event-augmented graph-LSTM architecture for abstractive summarization.

	Train		Val		Test	
	Source	Target	Source	Target	Source	Target
# pairs	5,178		500		999	
# tokens	878	437	873	434	894	444
# sentences	26	16	26	16	26	16
# words	643	348	643	347	653	352
# extracted events per document	2.63	—	2.54	—	2.70	—
# nodes per G_e/G_a	6/547	—	5/550	—	6/556	—

TABLE 4.1. CDSR dataset statistics. All values are mean except for “# pairs”. Sentence segmentation is made with spaCy [Honnibal et al., 2020].

4.2.4.4 Quantitative and Qualitative Evaluation

On the trails of common practice, we automatically evaluate model performance in terms of ROUGE-1, ROUGE-2, and ROUGE-L F1 scores. These recall-based metrics

Model	#params	R-1	R-2	R-L	Flesch-Kincaid	Coleman-Liau
ORACLE [†]	—	53.56	25.54	49.56	14.85	16.13
BERT-base [†]	110M	26.60	11.11	24.59	13.44	14.40
POINTER GENERATOR [†]	22M	38.33	14.11	35.81	16.36	15.90
BART-base (PubMed)	139M	51.20	19.77	48.47	13.69	13.45
BART-large (PubMed) [†]	406M	52.66	21.73	49.97	13.30	14.28
EASUMM [‡]	8M	46.30	18.73	43.78	12.42	13.06
COGITOERGOsumm	181M	52.23	<u>20.63</u>	49.44	14.10	13.67
- w/o RL	180M	<u>52.30</u>	20.47	<u>49.46</u>	14.06	13.64
- w/o event and RL	155M	52.13	20.42	49.30	14.02	13.69
- w/o AMR and RL	157M	52.02	20.54	49.25	13.97	13.66

TABLE 4.2. Automated evaluation on the full test set of CDSR with ROUGE (R in short) and readability metrics. Top: extractive models. Middle: abstractive models. Bottom: our semantics-augmented abstractive model. **Bold** and underline denote the best and second best R scores. [†] and [‡] results are from Guo et al. [2021b] and [Frisoni. et al., 2022], respectively. Our model significantly outperforms BART-base (Pitman’s permutation test, $p < 0.05$).

evaluate informativeness by measuring unigrams, bigrams, and longest common subsequence overlaps. A standard ROUGE metric does not shed meaningful light on other important dimensions like semantic coherence, abstractness, and intelligibility—which are focal for our work. To better gauge summary quality, we apply BERTScore [Zhang et al., 2020d], FactCC [Kryscinski et al., 2020], and report n-grams novelty⁷ in tandem with readability metrics, namely Flesch-Kincaid grade-level [Kincaid et al., 1975a] and Coleman-Liau [Coleman and Liau, 1975b] indices⁸. We utilize default hyperparameters for all the metrics except for BERTScore, where we use the IDF weighting and specify `rescale_with_baseline=True` to increase interpretability and avoid small range variations. In the quest to analyze our generated summaries qualitatively, we operate an in-depth human evaluation study. We randomly select 30 CDSR test set instances and invite 3 annotators—proficient English speakers with biomedical competencies—to access our models’ outputs, along with those of BART-base (presented in random order), i.e., the skeleton model oblivious to semantics. After reading the articles, each judge scores summaries on a Likert scale from 1 (worst) to 5 (best) in conformity with four independent perspectives: (i) *informativeness*, i.e., conveying salient content; (ii) *factuality*, i.e., being faithful to the article; (iii) *fluency*, i.e., being fluent, grammatical, and coherent; (iv) *succinctness*, i.e., non containing redundant and unnecessary information.

⁷Percentage of new word-level unigrams in the predicted summaries compared to the source [See et al., 2017].

⁸They estimate the years of education generally required to understand the summary. We compute them using <https://pypi.org/project/textstat/>. Lower scores indicate that the text is easier to read (U.S. college-level readability belongs to the range [13-16]).

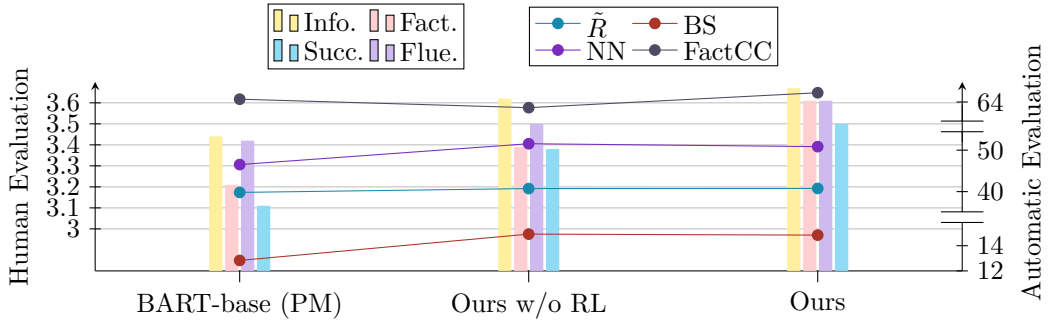


FIGURE 4.4. Human evaluation scores on **I**nformativeness, **F**actualness, **F**luency, **S**uccinctness, compared to the ROUGE-1/2/L average ($\tilde{R}$), BERTScore (BS), FactCC %, and average % novel n -grams w/ $n \in [1 - 4]$ (NN). COGITOERGOSUMM (ours) achieves significantly higher ratings than BART-base PubMed (student t-test, $p=0.0305$).

4.2.5 Results and Ablation Studies

Overall quantitative results are delighted in Table 4.2. Zooming out, we find that structured semantic information can greatly help a pre-trained language model recognize salient parts in source documents. Although our focus is on improving semantic consistency, the ROUGE scores attained by COGITOERGOSUMM are significantly higher than previous extractive and abstractive methods, except for BART-large, for which our models are still competitive despite having more than $2\times$ fewer trainable parameters. Remarkably, we beat graph-LSTMs by 6 R-1/-L points. The performance drops the most when removing graph encoders. Even if AMRs appear more impactful than events, the best results come from their mixture, indicating that the two types of semantic graphs complement each other in generating sounder summaries. RL effects are not appreciable by ROUGE, howbeit remarkable with a deeper analysis. Fig. 4.4 presents the human evaluation results contrasted to automatic metrics on the same sample. The average Kendall coefficient among all evaluators’ inter-rater agreement is 0.16. COGITOERGOSUMM ranks better on every quality dimension inspected, pronouncing the gap with BART (+12.46% factualness, +6.69% informativeness) and reaffirming previous deductions. The plot underlines the poor correlation between ROUGE and the desired output properties.

We validate the relative impact of our principal components (Table 4.3). Firstly, we scrutiny different graph encoders: (i) a GAT on Levi-transformed bipartite graphs treating nodes and edges equally, and (ii) an edge-aware GAT; we uncover that (ii) brings a substantial headroom that ascertains the value of relation types in representation learning. Secondly, we test different ways of combining event and AMR cross-attentions, documenting slightly better scores with a parallel strategy. Tests with different random seeds and extra qualitative case studies are disclosed in Appendix.

	Ablation	$\tilde{R}$	Readability	BS	NN
GNN Type	EGAT	40.74	15.17	14.88	51.55
	GAT bipartite	40.67	15.16	14.76	51.18
Attn. Comb.	Parallel	40.74	15.17	14.82	51.55
	Sequential (AMR, event)	40.65	15.08	14.98	51.81
	Sequential (event, AMR)	40.69	15.01	14.84	52.57

TABLE 4.3. Ablation results averaged over three runs. Evaluation on ROUGE-1/2/L average ($\tilde{R}$), Flesch-Kincaid and Coleman-Liau average (Readability), BERTScore (BS), and average % novel n -grams w/ $n \in [1 - 4]$ (NN).

4.2.6 Conclusion

In this thesis, we introduce a framework for infusing domain-specific and -general semantic parsing graphs—events and AMRs—into transformer-based models for biomedical abstractive summarization. We propose new decoder cross-attention modules and reward signals to generate high-quality summaries conditioned on both the source documents and their formal underlying semantics. Experiments and ablation studies on CDSR demonstrate that our framework sets new marks in informativeness, factuality, and readability, better selecting and preserving summary-worth content. Qualitative evaluation unveils that our models surpass current baselines on all metrics associated with human judgment while still being competitive on recall-based scores (i.e. ROUGE). Our results substantiate the hypothesis that semantic awareness through graph injection draws a complementary path to architectural scaling. For future work, we plan to model extracted semantics through logic representations so as to enable reasoning [Kapanipathi et al., 2021] and neuro-logic decoding [Lu et al., 2021].

4.3

Evidence, my Dear Watson: Abstractive Dialogue Summarization on Learnable Relevant Utterances

Originally published as: Paolo Italiani, Giacomo Frisoni, Gianluca Moro, Antonella Carbonaro, and Claudio Sartori. *Evidence, my dear watson: Abstractive dialogue summarization on learnable relevant utterances*. *Neurocomputing*, 572:127132, 2024

Abstract

Abstractive dialogue summarization requires distilling and rephrasing key information from noisy multi-speaker documents. Combining pre-trained language models with input augmentation techniques has recently led to significant research progress. However, existing solutions still struggle to select relevant chat segments, primarily relying on open-domain and unsupervised annotators not tailored to the actual needs of the sum-

marization task. In this thesis, we propose DEARWATSON, a task-aware utterance-level annotation framework for improving the effectiveness and interpretability of pre-trained dialogue summarization models. Precisely, we learn relevant utterances in the source document and mark them with special tags, that then act as supporting evidence for the generated summary. Quantitative experiments are conducted on two datasets made up of real-life messenger conversations. The results show that DEARWATSON allows model attention to focus on salient tokens, achieving new state-of-the-art results in three evaluation metrics, including semantic and factuality measures. Human evaluation proves the superiority of our solution in semantic consistency and recall. Finally, extensive ablation studies confirm each module’s importance, also exploring different annotation strategies and parameter-efficient fine-tuning of large generative language models.

4.3.1 Introduction

In today’s fast-paced world, dialogues have become ubiquitous and indispensable means of communication, with online chat applications (e.g., Whatsapp, Messenger, WeChat) being the most prominent case [Statista, 2022]. Automatic highlights can aid individuals and organizations in managing and comprehending the increasingly overwhelming amount of information exchange, ultimately supporting quick reviews and decision-making.

In the natural language generation (NLG) field, abstractive dialogue summarization is the task of producing a condensed and meaningful summary of a multi-speaker conversation that is not a verbatim representation of the original input. Compared to traditional writings, human-to-human dialogues pose unique challenges due to their dynamic, interactive, and first-person nature, often informal, prolix, and repetitive—peppered with false starts, backchanneling, speaker role shifting, reconfirmations, hesitations, and interruptions [Sacks et al., 1978]. Relevant content is often scattered across participants and dialogue turns, resulting in a lower information density and more diffuse topic coverage. These obstacles become even more pronounced when style and register are diversified, the setting is multi-party (>2 interlocutors), and sources reflect heterogeneous real-life subjects. Despite the recent strides achieved by pre-trained language models (PLMs), existing summarizers have *opaque behaviors* and pay attention to information pieces different from those included in the reference summary [Gliwa et al., 2019], undermining trustability.

The latest research has focused on input augmentation techniques to guide the generative model in identifying the most fundamental concepts, incorporating auxiliary annotations into the dialogue text. Prior studies have sought to automatically annotate keywords, redundant utterances, and topics [Feng et al., 2022]. Nevertheless, these annotations usually come from *burdensome human efforts* [Li et al., 2019b], *open-domain toolkits* [Zechner, 2002]—not suitable for dialogues, or *unsupervised strategies* [Murray et al., 2005, Liu et al., 2019b, Chen and Yang, 2020, Feng et al., 2021]—not designed to complement the task of interest, resulting in possible inconsistencies and information loss. The annotation process (gold or silver) is treated as a preprocessing step rather than a learning objective, capping potential benefits and generalization. Furthermore,

Srivastava et al. [Srivastava et al., 2023] have recently demonstrated the advantages of training or making zero-shot summary predictions exclusively with pertinent source sentences determined through fixed algorithms grounded on ROUGE-1 and topic segmentation. These advances raise expectations for learnable, reference-free, and dynamic annotation techniques, still lacking contributions.

We propose DEARWATSON, a novel task-aware utterance-level annotation framework designed to enhance the efficacy and interpretability of neural dialogue summarizers. Drawing on the knowledge encoded in PLMs, we train a classifier to detect the most relevant dialogue utterances for the summarization task. Predicted utterances are then wrapped with special tokens, and a pre-trained summarizer is fine-tuned on the augmented input. As depicted in Figure 4.5, we postulate that pointing out the meaningful source spans allows the generative model to better direct its attention to them; learning such annotations together with the summary can enhance final performance and unlock interpretability. We investigate two architectures following different learning schemes: (i) *joint*, where the annotator and the summarizer are treated as two independent units simultaneously trained, and the relevant target utterances come from a self-supervised heuristic; (ii) *end-to-end*, where the gradient backpropagates from the summarizer to the annotator, whose tag placement directly optimizes summary generation. Differentiable utterance selection is achieved via Gumbel-Softmax Trick [Jang et al., 2017b]. We run extensive experiments on two widely used dialogue summarization datasets, SAMSum [Gliwa et al., 2019] and DialogSum [Chen et al., 2021a], demonstrating that our models push the state-of-the-art according to five automatic evaluation metrics. Through ablation studies, we establish the significance of each module, ranging from the annotator implementation and underlying heuristic to the influence of the number of speakers and utterances. Qualitative experiments show a less diluted distribution of attention scores. Furthermore, human and ChatGPT-driven evaluations corroborate the evidence role of the selected utterances in making the summarizer behavior more transparent.

Our contributions can be summarized as follows:

- We pioneer input augmentation (summary-worthy utterance selection) as an additional training objective, without any reliance on human-crafted labels.
- We shed light on the overall efficacy of input augmentation in dialogue summarization, evaluating different annotation heuristics, and uncovering substantial room for future improvements.
- Beyond joint learning, we propose an architecture capable of making discrete choices without disrupting backpropagation. To this end, we suggest the incorporation of Gumbel-based sampling algorithms, a methodology previously unexplored in the context of evidence generation [Huijben et al., 2021]. Our research offers valuable insight into the intricate comparison between joint learning and end-to-end learning.
- We outperform all existing summarization baselines on two popular testbeds (up to +0.67/0.58/0.63 ROUGE-1/2/L F1 and +1.38 BERTScore, i.e., maximum Δ

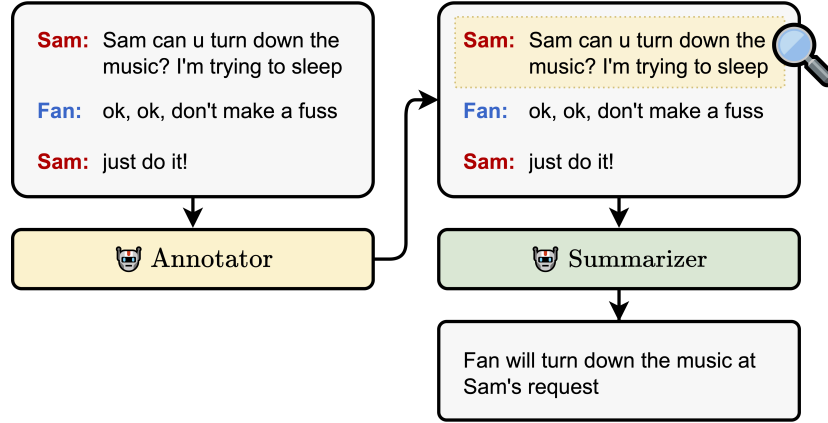


FIGURE 4.5. Overview of DEARWATSON. Utterance annotation and summarization are simultaneously mastered.

for each metric regardless of the dataset), and add interpretability with only minor incremental training/inference costs associated with the extra tagging task.

- We carry out a rigorous human evaluation to demonstrate that learned annotations are instrumental in understanding the dialogue segments that exert the greatest impact on a model prediction.
- We mark the first instance of harnessing ChatGPT for abstractive dialogue summarization, gauging its validity in terms of consistency with human judgment.

4.3.2 Related Work

Dialogue Summarization Pre-trained language models have sparked a paradigm shift in abstractive summarization [Zhang et al., 2020b, Lewis et al., 2020b], yielding unprecedented results even in multi-document [Moro et al., 2022b] and low-resource [Moro and Ragazzi, 2022b, Moro et al., 2023d] settings. Although most prior work addresses single-speaker content, dialogue summarization is gaining traction. Nevertheless, transferring conventional models to chats is not straightforward: avoiding overlooking essential information across turns is one of the primary requirements [Liu and Chen, 2021]. Since core knowledge is often fragmented and embodied in incomplete sentences, generating fluent summaries by utterance extraction alone is impractical. Popular approaches involve ad-hoc pretraining [Zhang et al., 2020e, Bao et al., 2020, Wu et al., 2020, Cao et al., 2020, Gao et al., 2020, Gu et al., 2021, Zhong et al., 2022b], while—following the steps of current trends [Frisoni et al., 2023a]—a complementary procedure is injecting structured data modeling conversation topics or dialogue acts [Chen and Yang, 2020, Liu et al., 2021]. In this thesis, we enrich the dialogue text to enhance the effectiveness and interpretability of any summarizer.

Annotation for Abstractive Summarization Conceptually, Peyrard [Peyrard, 2019] posits that a high-quality summary is closely tied to three dimensions: informativeness, redundancy, and relevance. Past research has actively accounted for these aspects by incorporating auxiliary information into the dialogue. To bolster informativeness, some work tagged linguistically-grounded keyphrases [Riedhammer et al., 2008], domain terminologies [Koay et al., 2020], and topic words [Zhao et al., 2020]. To mitigate redundancy, the authors labeled repetitive utterances with similarity-based methods [Zechner, 2002, Murray et al., 2005]. To maximize relevance, other groups carried out topic segmentation [Li et al., 2019b, Liu et al., 2019b, Chen and Yang, 2020]. Feng et al. [Feng et al., 2021] drawn support from forward passing on a pre-trained conversational response generation model to tag the input text based on loss scores and utterance embeddings. Input augmentation has also been exploited to control text generation, entailing the introduction of auxiliary signals to enforce the desired output properties [Wu et al., 2021a, Liu and Chen, 2021, Frisoni et al., 2022b]. All such annotations are procured through manual labor, pre-defined heuristics, open-domain toolkits, or unsupervised techniques. Instead, we conjecture that model performance remains subpar when treating input augmentation outside of learning objectives. To our knowledge, we are the first to optimize the annotator in tandem with the summarization task.

4.3.3 Preliminaries

4.3.3.1 Task Definition

Abstractive dialogue summarization endeavors to craft a creative summary $\mathcal{S}$ of $|\mathcal{S}|$ tokens $[s_1, s_2, \dots, s_{|\mathcal{S}|}]$ from a dialogue $\mathcal{D}$ consisting of $|\mathcal{D}|$ utterances $[\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{|\mathcal{D}|}]$. $[\cdot]$ denotes concatenation. Each utterance $\mathbf{u}_i$ comprises $|\mathbf{u}_i|$ tokens $[\mathbf{p}_i, u_{i,1}, u_{i,2}, \dots, u_{i,|\mathbf{u}_i|}, \text{SEP}_i]$, where $i \in [1, |\mathcal{D}|]$, $\mathbf{p}_i$ is the speaker, and SEP_i signifies the utterance end. Therefore, the task can be formalized as producing the summary $\mathcal{S}$ given the input: $\mathcal{D} = [\mathbf{p}_1, u_{1,1}, \dots, \text{SEP}_1, \dots, \mathbf{p}_{|\mathcal{D}|}, u_{|\mathcal{D}|,1}, \dots, \text{SEP}_{|\mathcal{D}|}]$.

4.3.3.2 Oracle Dialogue Annotation

Chat data is rife with irrelevant content, such as greetings and inconsequential turns, that does not contribute to producing an informative resume. Hence, steering the model’s attention toward core utterances may bear substantial opportunities. Gold utterance relevance labels for a dialogue summarization task are generally not available, and acquiring them would be a costly endeavor. To make our solution adaptable and effective in real-world scenarios, we have devised multiple self-supervised heuristics. Taking a closer look, we propose three alternative approaches to designate a source utterance $\mathbf{u}_i$ as relevant based on its *overlap* ($\cap$) with the gold summary reference.

- **ONE-WORD-OVERLAP:** inspired by Wu et al. [Wu et al., 2021a], $\mathbf{u}_i$ must feature at least one word (other than stopwords) from the target summary.
- **TOP-P-WORD-OVERLAP:** more severely, $\mathbf{u}_i$ must rank within the top $p\%$ of the

source utterances that exhibit the highest similarity to the target summary, measured using ROUGE-1.

- **TOP-P-SEMANTIC-OVERLAP:** $\text{---}\text{u}\text{---}$ measured using cosine similarity. To accomplish this, we leverage a state-of-the-art Sentence Transformer,⁹ computing the normalized dot product between semantically-informed utterance embeddings.

Please note that, unlike ONE-WORD-OVERLAP, the hyperparameter p offers control over the annotation percentage. According to the relevance criterion under investigation, we flag core utterances by inserting the special tokens $\langle \mathbf{h1} \rangle$ and $\langle \backslash \mathbf{h1} \rangle$ at their beginning and end, respectively. Equation 4.15 succinctly outlines the utterance map function for producing the annotated dialogue $\mathcal{D}^*$.

$$\text{ann}(\mathbf{u}_i) = \begin{cases} [\langle \mathbf{h1} \rangle_i, \mathbf{p}_i, \dots, \langle \backslash \mathbf{h1} \rangle_i, \text{SEP}_i] & \mathbf{u}_i \cap \mathcal{S} \neq \emptyset \\ \mathbf{u}_i & \mathbf{u}_i \cap \mathcal{S} = \emptyset. \end{cases} \quad (4.15)$$

We note that this is only possible during training, since we obviously do not have access to the target summary at inference time. Empirical tests from our group exhibit that PLMs fine-tuned on oracle $\mathcal{D}^*$, rather than $\mathcal{D}$, gain up to 4.5 ROUGE-1/2/L average points (see subsection 4.3.6.1 for details). These results attest to the great controllability power of the utterance annotation strategy and motivate the architectures discussed in subsection 4.3.4, where we naturally eliminate the prior target knowledge assumption to define general-purpose *annotate-then-summarize* models.

4.3.4 Method

This subsection will introduce two supervised architectures for highlighting relevant utterances and condensing the augmented dialogue. First, we will explore a *joint* paradigm, breaking down the goal into two subtasks learned simultaneously (subsection 4.3.4.1). Then we will move to an *end-to-end* framework, taking annotation and summarization as a single unit by differentiable input augmentation (subsection 4.3.4.2). We sketch both solutions in Figure 4.6.

4.3.4.1 Joint Learning Paradigm

Utterance Classification

We fine-tune a PLM to predict relevant utterances, i.e., dialogue lines mentioning facts expected to appear in the resume. To achieve this goal, the $\text{ann}(\mathbf{u}_i)$ heuristic disclosed in subsection 4.3.3.2 is treated as the source of ground-truth annotations. Technically, we supply $\mathcal{D}$ to a bidirectional text encoder $E_t(\cdot)$, deriving a contextual hidden representation $\mathbb{R}^d$ for each dialogue token:

$$[\mathbf{h}_{\mathbf{p}_1}, \mathbf{h}_{u_{1,1}}, \dots, \mathbf{h}_{\text{SEP}_{|\mathcal{D}|}}] = E_t(\mathcal{D}). \quad (4.16)$$

⁹<https://huggingface.co/sentence-transformers>

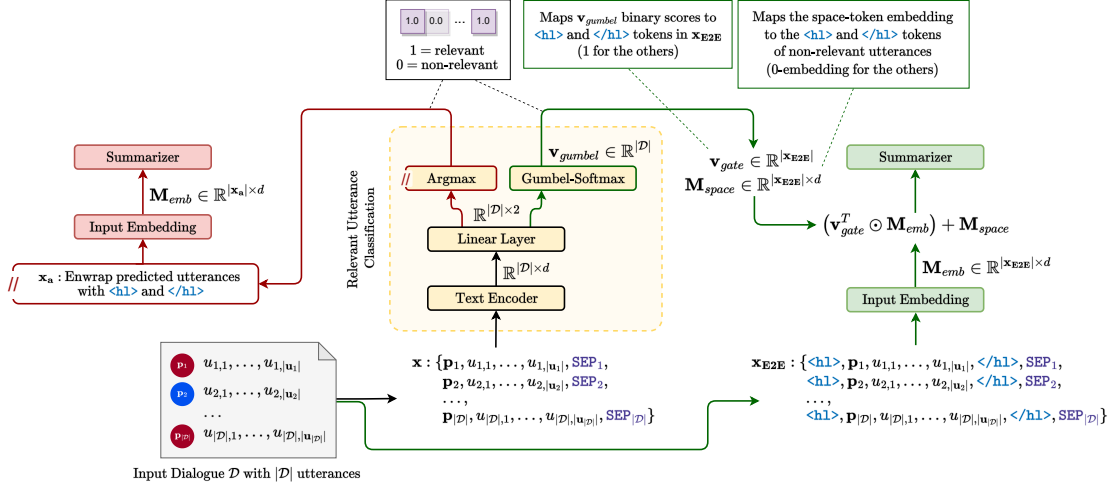


FIGURE 4.6. Illustration of our joint and end-to-end DEARWATSON architectures. Following a shared initial phase for utterance classification, joint learning peculiarities are highlighted in red, while the alternative end-to-end modifications are shown in green. The “//” symbol connotes backpropagation interruption points.

We extend the model training to represent each utterance in the corresponding SEP token. Indeed, on top of the encoder stack, we add a task-specific linear layer to project each utterance embedding $\mathbf{h}_{\text{SEP}_i}$ into two-dimensional unnormalized logits $\mathbf{l}_i$ (confidences for relevant T and non-relevant F classes):

$$[\mathbf{l}_1, \dots, \mathbf{l}_{|\mathcal{D}|}] = \mathbf{W}_c \{\mathbf{h}_{\text{SEP}_1}, \dots, \mathbf{h}_{\text{SEP}_{|\mathcal{D}|}}\} + \mathbf{b}_c, \quad (4.17)$$

where $\mathbf{W}_c$ represent trainable weights and $\mathbf{b}_c$ are bias parameters. We optimize a binary cross-entropy loss $\mathcal{L}_{ce}$:

$$\mathcal{L}_{ce} = - \sum_i^{\{T, F\}} y_i \log(p_i), \quad (4.18)$$

where y_i is the truth class label and p_i is the softmax probability on the output of Equation 4.17 for the i^{th} class.

Summarization

We unveil relevant utterances by applying the argmax function to the output of the classification head, wrapping them with special $\langle h1 \rangle \langle /h1 \rangle$ tokens. Finally, we feed the annotated dialogue $\mathcal{D}^*$ to a pre-trained transformer-based summarizer. To maximize the estimated probability P_θ of the actual summary $\mathcal{S}$, we utilize a negative log-likelihood loss function:

$$\mathcal{L}_{nll} = - \sum_i \log P_\theta(s_i | \mathcal{D}^*). \quad (4.19)$$

The classifier and the summarizer are jointly trained by minimizing $\mathcal{L} = \mathcal{L}_{ce} + \mathcal{L}_{nll}$.

4.3.4.2 End-to-End Paradigm

The utterance annotation process described in subsection 4.3.4.1 halts the gradient flow from the summarizer to the classifier. Non-differentiability is caused by the argmax step function, which outputs discrete class labels, i.e., hard assignments. When concerned with discrete stochasticity, challenges arise regarding both sampling from discrete distributions and gradient estimation thereof. Reinforcement learning-free approaches often relax sampling functions by employing continuous approximations, such as Soft-argmax [Zhang et al., 2017] or Gumbel-Softmax [Jang et al., 2017a]. Gumbel-based sampling algorithms, in particular, have found success in a range of applications, such as generating text through stochastic beam search for dialogue systems [Firdaus et al., 2020] and machine translation [Kool et al., 2019], as well as leaning communication protocols in multi-agent games [Havrylov and Titov, 2017]. In this thesis, our proposed end-to-end architecture incorporates the Gumbel-Softmax Trick for categorical reparameterization.

Gumbel-Softmax Trick Gumbel-Softmax is an efficient gradient estimator for sampling from a categorical distribution, which unlocks backpropagation through the entire network even when intermediate outputs are discrete—directly mapping the raw dialogue to the summary on the augmented input. Let c be our categorical variable with probabilities $\{\pi_T, \pi_F\}$. Categorical samples are encoded as two-dimensional one-hot vectors at the corners of a mono-dimensional simplex Δ^1 . Utterance relevance choices from our $|\mathcal{D}|$ -length c distribution are estimated by generating $z \in \Delta^1$:

$$z_i = \frac{\exp((\log(\pi_i) + g_i)/\tau)}{\sum_j^{\{T,F\}} \exp((\log(\pi_j) + g_j)/\tau)} \quad \text{for } i \in \{T, F\}, \quad (4.20)$$

where τ is the softmax temperature and g_T, g_F are i.i.d. noise samples drawn from $\text{Gumbel}(0, 1)$ by pulling out $u \sim \text{Uniform}(0, 1)$ and then computing $g = -\log(-\log(u))$. At lower temperatures ($\tau \rightarrow 0$), z samples are identical to those from a categorical distribution; at higher temperatures ($\tau \rightarrow \infty$), z samples are no longer one-hot and become uniform. The Gumbel-Softmax distribution is smooth for $\tau > 0$, thus having a well-defined gradient. While z samples are differentiable, our annotation process requires a discrete selection. Hence, we adopt the Straight-Through (ST) Gumbel-Softmax, which discretizes z through argmax in the forward pass and utilizes the continuous approximation in the backward pass. using the ST Gumbel-Softmax on $[\mathbf{1}_1, \dots, \mathbf{1}_{|\mathcal{D}|}]$, we derive a one-hot matrix $\mathbf{M}_{gumbel} \in \mathbb{R}^{|\mathcal{D}| \times 2}$, where each utterance is assigned the vector $[0.0, 1.0]$ if relevant and $[1.0, 0.0]$ otherwise. A vectorial representation $\mathbf{v}_{gumbel} \in \mathbb{R}^{|\mathcal{D}|}$ is $\mathbf{M}_{gumbel}(:, 2)$.

Annotation Gate To preserve the gradient flow after classification, it is necessary to identify a differentiable strategy also for dialogue augmentation. Compared to joint learning, we (i) reframe the input as $\mathbf{x}_{\mathbf{E2E}}$ by positioning the $\langle \mathbf{h1} \rangle$ tokens at the start and end of each utterance, (ii) delineate a gating mechanism to dynamically turn off the wrapping annotation of unselected utterances. If a dialogue line is deemed irrelevant,

# Dialogues	# Participants			# Turns			# Dialogue Words			# Summary Words			
	Mean	Std	Range	Mean	Std	Range	Mean	Std	Range	Mean	Std	Range	
SamSum													
Train	12,460	2.40	0.83	[1, 14]	11.17	6.45	[1, 46]	124.5	94.2	[13, 1017]	23.44	12.72	[2, 73]
Dev	818	2.39	0.84	[2, 12]	10.83	6.37	[3, 30]	121.6	94.6	[18, 691]	23.42	12.71	[4, 68]
Test	819	2.36	0.83	[2, 11]	11.25	6.35	[3, 30]	126.7	95.7	[17, 669]	23.12	12.20	[4, 71]
DialogSum													
Train	12,460	2.01	0.13	[2, 7]	9.49	4.16	[2, 61]	197	97.83	[52, 1389]	31.02	13.51	[7, 212]
Dev	500	2.01	0.13	[2, 4]	9.38	3.99	[2, 29]	194.38	90.81	[55, 672]	28.96	12.63	[7, 89]
Test	500	2.01	0.09	[2, 3]	9.71	4.99	[2, 65]	202.22	108.84	[54, 1258]	25.59	11.01	[6, 96]

TABLE 4.4. SAMSum and DialogSum dataset statistics. Word counts are determined by the NLTK tokenizer [Bird et al.].

we replace the embeddings of its $\langle \text{h1} \rangle / \langle \text{h1} \rangle$ tokens with those of space tokens (i.e., " "), mimicking their absence in the input. Mechanically, we obtain the representation of each token $\mathbf{M}_{emb} \in \mathbb{R}^{|\mathbf{x}_{\mathbf{E2E}}| \times d}$ by feeding $\mathbf{x}_{\mathbf{E2E}}$ to the summarizer’s input embedding layer. We construct a mask $\mathbf{v}_{gate} \in \mathbb{R}^{|\mathbf{x}_{\mathbf{E2E}}|}$ to cancel undesired special tags, where the binary weight of each token $t_i \in \mathbf{x}_{\mathbf{E2E}}$ is established with the following function:

$$\text{mask}(t_i) = \begin{cases} \mathbf{v}_{gumbel}[\text{utterance}(t_i)] & t_i \in \{\langle \text{h1} \rangle, \langle \text{h1} \rangle\} \\ 1 & t_i \notin \{\langle \text{h1} \rangle, \langle \text{h1} \rangle\}. \end{cases} \quad (4.21)$$

We build a matrix $\mathbf{M}_{space} \in \mathbb{R}^{|\mathbf{x}_{\mathbf{E2E}}| \times d}$ containing space-token embeddings for annotation tags of irrelevant utterances, 0-embeddings otherwise. The matrix supplied to the summarizer is $(\mathbf{v}_{gate}^T \odot \mathbf{M}_{emb}) + \mathbf{M}_{space}$, while the final loss is $\mathcal{L} = \mathcal{L}_{ce} + \mathcal{L}_{nll}$ as in subsection 4.3.4.1.

4.3.5 Experimental Setup

Implementation details, computing infrastructure, and experiment hyperparameters are described in Appendix A. For all the models reported in subsection 4.3.5.2, we adopt the optimal configurations indicated by the authors, tailored to our hardware capacity.

4.3.5.1 Dataset

On the steps of previous research, we carry out experiments on SAMSum [Gliwa et al., 2019] and DialogSum [Chen et al., 2021b], two modern testbeds for abstractive dialogue summarization in the English language. SAMSum is prepared by Samsung R&D Institute Poland. Linguists created and wrote conversations to reflect the complexity and proportion of daily topics. DialogSum is gathered from a practice website and three publicly available dialogue datasets, namely Dailymail [Li et al., 2017], DREAM [Sun et al., 2019], MuTual [Cui et al., 2020]. For both of these two benchmarks, the style and register are diversified (informal, semi-formal, formal), encompassing a broad spectrum of everyday subjects, such as education, employment, healthcare, chit-chats, meeting organization, shopping, recreation, and travel. Summaries are presented in the third person and offer succinct overviews of the discussed topics. Datasets statistics are detailed in Table 4.4.

4.3.5.2 Models

Classification and Summarization Modules Our annotate-then-summarize framework is agnostic to the underlying models. We investigate several transformer-based PLMs with different capacities (i.e., architectures, pretraining schemes), aiming to reuse their linguistic and semantic knowledge. We also fine-tune large language models (LLMs) with 1B parameters. As for the classifier, we test representation and generative networks to encode each utterance’s last token (SEP). Specifically, we put into play ROBERTA [Liu et al., 2019a], DEBERTAV2 [He et al., 2021], GPT-2 [Radford et al., 2019a], and OPT [Zhang et al., 2022]. For TOP-P-SEMANTIC-OVERLAP, semantic similarity is calculated with a ROBERTA-large model fine-tuned on natural language inference (NLI).¹⁰ As for the summarizer, we evaluate BART [Lewis et al., 2020b]—the most popular generative model for dialogue-oriented tasks [Feng et al., 2022], rooted in denoising pre-training objectives, and FLAN-T5 [Chung et al., 2022b]—a large-scale model fine-tuned with instructions on a mixture of text-to-text tasks (dialogue summarization excluded).

Baselines We head-to-head compare our DEARWATSON models with a plethora of competitive abstractive summarization baselines.

- BART, FLAN-T5. Vanilla models fine-tuned on non-augmented dialogues, i.e., without our learnable annotation.
- MV-BART [Chen and Yang, 2020]. BART model with multi-view decoder attention layers to incorporate conversation topics and progression stages.
- COREF-ATTN [Liu et al., 2021]. BART model with an additional coreference-guided attention layer between the encoder and the decoder.
- S-BART [Chen and Yang, 2021a]. BART model with a multi-granularity decoder incorporating utterance dependency graphs and action graphs.
- BART-DIALOGPTANN [Feng et al., 2021]. BART model trained on documents augmented with unsupervised topic, keyword, and redundancy annotations from DialoGPT [Zhang et al., 2020e].
- SWING [Huang et al., 2023b]. BART model trained with NLI signals to encourage coverage and factuality.
- DIALSENT [Jia et al., 2022]. BART model post-trained to rephrase from dialogues to narratives before fine-tuning for dialogue summarization.

The large version is harnessed across each BART-derived model to promote fairness.

4.3.5.3 Metrics

We quantify classification effectiveness with recall, precision, F1-score, and accuracy for positive utterance classification T . Following the trail of common practice, we evaluate summarization performance in terms of ROUGE-1/2/L F1 scores [Lin, 2004]. Inspired by Moro et al. [Moro et al., 2023c], we also measure an aggregated ROUGE judg-

¹⁰<https://huggingface.co/sentence-transformers/roberta-large-nli-stsb-mean-tokens>

ment: $\mathcal{R} = \text{avg}(r_1, r_2, r_L) / (1 + \sigma_r^2)$, where $r_{1/2/L} \in [0, 1]$ and σ_r^2 is the F1 variance.¹¹ To refine summary quality assessment and go beyond lexical superficiality, we make use of BERTScore [Zhang et al., 2020d] and BARTScore [Yuan et al., 2021b], two recently-developed model-based metrics that have been shown to correlate well with human judgments (i.e., semantic coverage, coherence, informativeness, relevance, fluency, and factual consistency dimensions). Higher scores indicate better overall results. Metric settings and interpretability hints are documented in Appendix A.2.

4.3.6 Results

4.3.6.1 Quantitative Evaluation

Utterance Classification Table 4.5 displays preliminary fine-tuning results about the influence of the PLM choice on the classification component alone. The reported results are based on the most effective annotation strategy identified for each dataset in the downstream summarization task. A thorough exploration of the strategy’s impact is detailed in the next paragraph. Notably, RoBERTa stands out from the other models, exhibiting the highest F1 and accuracy scores, despite having up to 4.5x fewer parameters. Accordingly, we opt for RoBERTa as the classification backbone in all our complete solutions, denoted by DW_{JL} (joint learning) and DW_{EE} (end-to-end).

	Recall	Precision	F1	Accuracy
SAMSum				
GPT-2-XL	<u>86.11</u>	75.97	80.72	80.43
OPT-1.3B	84.10	<u>77.78</u>	<u>80.82</u>	<u>81.00</u>
DeBERTaV2-xlarge	87.34	74.27	80.27	79.60
RoBERTa-large	82.15	79.90	81.01	81.70
RoBERTa-large(DW_{JL})	83.03 $\uparrow$	<u>78.68</u>	<u>80.79</u>	<u>81.24</u>
RoBERTa-large(DW_{EE})	85.00 $\uparrow$	78.43	81.59 $\uparrow$	81.77 $\uparrow$
DialogSum				
GPT-2-XL	<u>67.67</u>	63.35	65.44	70.17
OPT-1.3B	67.16	65.02	66.07	71.22
DeBERTaV2-xlarge	65.57	62.18	63.83	69.07
RoBERTa-large	69.33	68.95	69.14	74.24
RoBERTa-large(DW_{JL})	61.38	66.67	63.91	73.30
RoBERTa-large(DW_{EE})	67.47	<u>67.23</u>	<u>67.35</u>	<u>73.95</u>

TABLE 4.5. Relevant utterance classification results on SAMSum and DialogSum test sets (3-runs average). All scores correspond to the annotation strategy that has been empirically identified as the most effective for each dataset, specifically, ONE-WORD-OVERLAP for SAMSum and TOP-P-WORD-OVERLAP for DialogSum. Each dataset’s results are categorized into two subsections. Top: classification-only fine-tuning; bold and underline denote best and second-best scores. Bottom: annotate-then-summarize fine-tuning; $\uparrow$ highlights relative improvements.

¹¹ $\mathcal{R}$ penalizes model results with discrepant unigram, bigram, and longest common subsequence overlaps.

Annotation Strategy The way "relevance" is defined for source utterances can have a profound impact on the quality of the generated summary. Table 4.6 highlights how the choice of different annotation strategies directly affects summarization metrics. Interestingly, the optimal strategy appears to be highly dependent on the dataset at hand, revealing contrasting preferences between SAMSum and DialogSum. To be specific, ONE-WORD-OVERLAP appears to be the best choice for SAMSum and the worst for DialogSum. Given the similar nature of the two benchmarks, we posit that a pivotal factor influencing the ultimate performance lies in achieving a harmonious balance between annotated and non-annotated oracle utterances, essentially the equilibrium between positive and negative classification examples. To go into this interplay, we present the percentage of annotated utterances for each strategy-dataset combination. For the top- p heuristics, the annotation percentage is equal to p ; we report the best value registered after a grid search in the hyperparameter space (see Table 4.13 for details). Adding weight to our hypothesis is the optimal value of p , which closely approaches 42%. Simultaneously, in SAMSum and DialogSum, ONE-WORD-OVERLAP annotates approximately $\approx 48\%$ and $\approx 64\%$ of the source utterances, respectively, indicating a greater likelihood of false positive relevance labels in the latter. Regarding the evaluation of TOP-P-WORD-OVERLAP and TOP-P-SEMANTIC-OVERLAP in DialogSum, our findings show that the former excels in joint learning and secures higher performance rankings in end-to-end training. As a result, we opt for TOP-P-WORD-OVERLAP.

	ROUGE-F	BERTScore-F	BARTScore-F	Ann. Utterances
SAMSum				
BART(DW_{JL}) · ONE-WORD-OVERLAP	44.50	<u>53.61</u>	-2.784	47.59%
BART(DW_{E2E}) · ONE-WORD-OVERLAP	<u>44.33</u>	53.72	<u>-2.783</u>	
BART($\bar{D}\bar{W}_{JL}$) · TOP-P-WORD-OVERLAP	43.91	53.31	<u>-2.787</u>	
BART(DW_{E2E}) · TOP-P-WORD-OVERLAP	43.52	53.07	-2.778	
BART($\bar{D}\bar{W}_{JL}$) · TOP-P-SEMANTIC-OVERLAP	43.79	53.29	-2.799	
BART(DW_{E2E}) · TOP-P-SEMANTIC-OVERLAP	43.86	53.06	-2.798	
DialogSum				
BART(DW_{JL}) · ONE-WORD-OVERLAP	37.36	51.19	-2.841	63.67%
BART(DW_{E2E}) · ONE-WORD-OVERLAP	37.33	51.40	-2.842	
BART($\bar{D}\bar{W}_{JL}$) · TOP-P-WORD-OVERLAP	38.29	52.45	-2.847	
BART(DW_{E2E}) · TOP-P-WORD-OVERLAP	<u>38.60</u>	52.83	<u>-2.826</u>	
BART($\bar{D}\bar{W}_{JL}$) · TOP-P-SEMANTIC-OVERLAP	37.59	51.76	-2.822	
BART(DW_{E2E}) · TOP-P-SEMANTIC-OVERLAP	38.66	<u>52.78</u>	-2.834	

TABLE 4.6. Quantitative summarization results on the SAMSum and DialogSum test sets after fine-tuning with different target annotation strategies. Oracle-annotated utterances are specified on the right. Bold and underline denote the best and second-best scores for each dataset.

Dialogue Summarization Overall results are delighted in Table 4.7, considering the optimal annotation strategy identified for each dataset. The remarkable outcomes produced by the oracle utterance annotations (up to +5.60/3.71/4.34 ROUGE-1/2/L F1, +4.68 BERTScore F1, +0.132 BARTScore F1) clearly affirm the powerful effect

of `<h1></h1>` tagging and set out a promising theoretical upper bound (i.e., perfect classifier with 0 false positives and negatives). Expressly, the annotation confers more benefits as the number of parameters decreases, reaching a peak with BART. When evaluated on SAMSum, our DEARWATSON models outmatch vanilla summarizers in almost all dominant metrics, especially precision measures. Moving to DialogSum, a BART-based backbone becomes sufficient to get competitive or superior ROUGE F1 scores, while greatly improving semantic and factual metrics (up to +13.82 BERTScore F1 and +0.871 BARTScore F1 compared to SWING and DIALSENT). In the context of ROUGE metrics, making a definitive choice between DW_{JL} and DW_{EE} proves challenging, as the two training modes exhibit distinct performance gaps in SAMSum and DialogSum. DW_{JL} is more robust and predictable, attributed to its finer error containment, which becomes increasingly challenging as the architecture grows. On the other hand, DW_{EE} consistently favor semantic metrics. In addition, DEARWATSON multi-task learning positively affects the prediction of relevant utterances, further enhancing ROBERTA capabilities; this becomes particularly evident when promoting more interaction through DW_{EE} . Our FLAN-T5(DW_{JL}) eclipses MV-BART on SAMSum and clearly sets a new state-of-the-art, beating multi-view, coreference-centered, multi-modal, and NLI-based alternatives.

Model	ROUGE-1			ROUGE-2			ROUGE-L			BERTScore			BARTScore		
	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F
SAMSum															
MV-BART	57.51	55.85	54.05	30.74	29.49	28.56	53.16	51.54	50.57	53.90	53.46	53.64	-2.915	-2.661	-2.788
COREF-ATTN	56.61	57.12	53.93	29.79	30.68	28.58	52.26	52.49	50.39	53.32	<u>53.89</u>	53.56	-2.915	-2.674	-2.794
S-BART	51.22	56.00	50.70	25.84	28.12	25.50	48.17	51.87	48.08	48.86	52.36	50.57	-3.110	-2.895	-3.003
BART-DIALOGPTANN	54.90	<u>57.90</u>	53.70	29.53	31.22	28.79	51.54	<u>53.72</u>	50.81	50.48	42.95	46.66	-2.975	-3.617	-3.296
SWING	57.19	55.27	53.04	30.34	29.99	28.36	52.91	51.59	50.08	53.75	52.49	53.07	-2.906	-2.728	-2.817
DIALSENT	55.68	57.26	53.54	30.05	31.14	28.91	51.55	52.82	50.21	52.86	53.92	53.34	-2.95	-2.722	-2.836
BART	<u>58.08</u>	53.93	53.06	30.66	28.74	28.08	53.02	49.83	49.44	54.06	51.52	52.74	-2.901	-2.726	-2.813
<i>Ours</i> · BART(DW_{oracle})	60.73	60.34	58.64	32.70	32.86	31.79	55.04	54.97	53.78	57.44	57.41	57.42	-2.788	-2.574	-2.681
<i>Ours</i> · BART(DW_{JL}) †	57.76	<u>55.79</u>	53.92	31.22	<u>30.41</u>	29.16	53.16	<u>51.58</u>	50.43	54.27	<u>53.04</u>	53.61	-2.885	-2.684	-2.784
<i>Ours</i> · BART(DW_{EE}) †	57.80	55.49	53.75	30.80	<u>30.12</u>	28.84	53.23	51.47	50.40	54.39	53.14	53.72	-2.891	-2.675	-2.783
FLAN-T5	58.36	56.59	<u>54.38</u>	32.36	<u>31.46</u>	<u>30.05</u>	54.15	52.79	<u>51.35</u>	55.10	53.73	<u>54.36</u>	-2.822	-2.668	-2.745
<i>Ours</i> · FLAN-T5(DW_{oracle})	59.45	58.12	56.22	33.15	32.55	31.31	55.33	54.32	53.09	56.69	55.65	56.13	-2.788	-2.612	-2.700
<i>Ours</i> · FLAN-T5(DW_{JL}) †	57.90	58.09	55.05	<u>32.26</u>	32.56	30.63	53.89	54.08	51.98	55.10	54.82	54.91	<u>-2.827</u>	-2.646	-2.737
<i>Ours</i> · FLAN-T5(DW_{EE}) †	57.94	56.57	54.13	32.14	31.33	29.86	<u>53.99</u>	52.87	51.28	<u>54.74</u>	53.37	54.00	-2.828	<u>-2.659</u>	<u>-2.744</u>
DialogSum															
SWING	54.47	44.85	47.67	24.99	20.82	21.99	50.79	43.28	45.72	45.81	41.04	43.45	-3.576	-3.343	-3.459
DIALSENT	46.46	52.16	47.60	21.05	24.34	21.76	44.60	49.26	45.79	36.61	41.36	39.01	-3.760	-3.633	-3.697
BART	53.37	43.83	46.55	23.87	20.00	21.00	49.82	42.32	44.70	54.59	<u>48.33</u>	<u>51.45</u>	<u>-2.865</u>	<u>-2.824</u>	<u>-2.844</u>
<i>Ours</i> · BART(DW_{oracle})	58.67	48.01	51.24	27.50	22.95	24.27	53.60	45.38	48.13	57.61	51.09	54.34	-2.741	-2.730	-2.735
<i>Ours</i> · BART(DW_{JL}) †	<u>53.78</u>	<u>45.47</u>	<u>47.79</u>	<u>24.05</u>	<u>20.84</u>	21.62	<u>49.93</u>	43.45	45.47	55.23	43.45	45.47	-2.861	-2.833	-2.847
<i>Ours</i> · BART(DW_{EE}) †	52.53	<u>47.38</u>	48.18	23.70	<u>21.79</u>	<u>21.95</u>	48.84	<u>44.82</u>	<u>45.66</u>	<u>54.67</u>	50.97	52.83	-2.867	-2.785	-2.826

TABLE 4.7. Quantitative summarization results on the SAMSum and DialogSum test sets after fine-tuning (3-runs average). Top: abstractive baselines. Bottom: backbone vanilla models and our annotate-then-summarize variants, including oracle utterance annotations, joint learning, and end-to-end learning. Bold and underline denote the best and second-best scores, excluding target-aware oracle results. The green gradient spotlights our relative percentage improvement compared to backbone models (the deeper, the more); † symbolizes statistical significance (Pitman’s permutation test, $p < 0.05$).

Annotation Statistics Table 4.8 wraps up fine-grained and coarse-grained annotation coverage statistics in augmented dialogue. On average, models trained in a joint learning mode are more likely to label short utterances.

Dataset	Model	Ann. Utterances	Ann. Words	Gen. Words	Gen. Tokens
SAMSum	BART	50.14	65.3	25.7	28.62
	BART(DW_{JL})	51.49	67.05	24.74	27.66
	BART(DW_{E2E})	51.49	67.05	25.0	27.95
	FLAN-T5	52.58	67.97	24.97	30.94
	FLAN-T5(DW_{JL})	52.58	67.97	23.92	29.59
	FLAN-T5(DW_{E2E})	49.92	64.85	24.79	30.76
DialogSum	BART	35.47	43.88	30.91	34.66
	BART(DW_{JL})	51.49	67.05	29.66	33.48
	BART(DW_{E2E})	39.97	47.99	27.99	31.68

TABLE 4.8. Average relevant utterance annotation and generation statistics. % of annotated utterances out of the total. % of annotated words out of the total. Number of generated words. Number of generated tokens.

Impact of Speakers and Utterances Figure 4.7 breaks down the $\mathcal{R}$ effectiveness by number of participants and utterances. Please note that DialogSum only has 2 or 3 speakers (see Table 4.4). Irrespective of the model or learning approach, as the number of utterances increases, ROUGE scores gradually decline in SAMSum. In contrast, in DialogSum, they exhibit a less predictable pattern, characterized by alternating spikes of good and poor performance. Importantly, SAMSum dialogues with 5+ speakers are accompanied by a marked $\mathcal{R}$ reduction. We exclude speaker number classes with fewer than three occurrences.

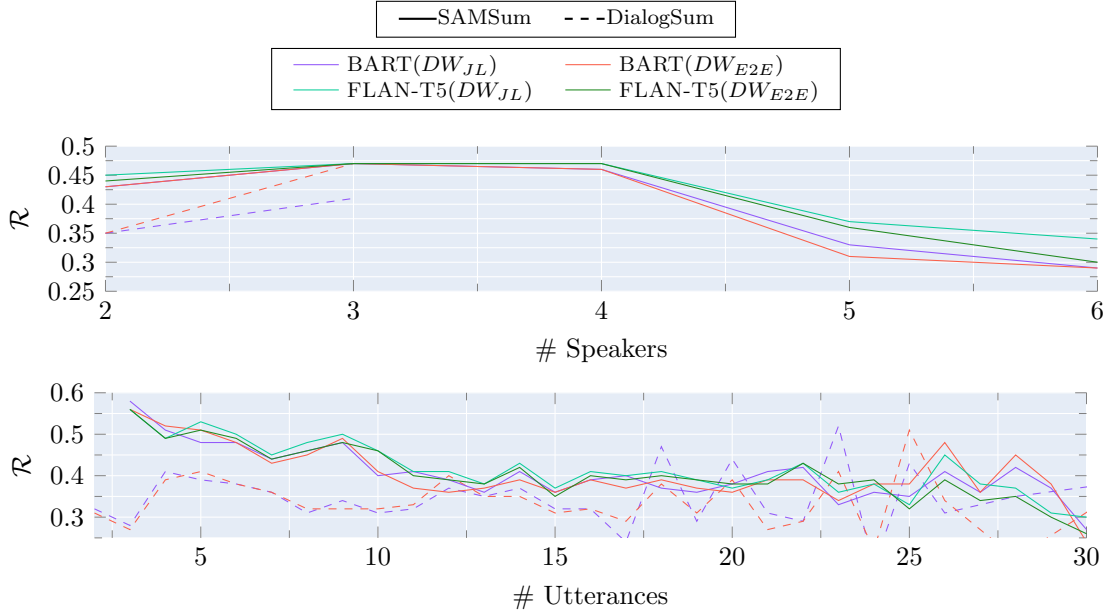


FIGURE 4.7. Relation between overall summarization ROUGE scores ($\mathcal{R}$) and the number of speaker-s/utterances in conversations.

4.3.6.2 Cross-Attention Analysis and Interpretability

We analyze the cross-attention values between the input dialogue and the generated summary to explicate the performance boost permitted by relevant utterance annotations. Figure 4.8 contrasts the behavior of BART and BART(DW_{JL}) on a qualitative SAMSum example. The standard BART model places little focus on salient dialogue tokens such as [bring, piece, later, on], causing essential facts to be missing from the resume. In BART(DW_{JL}), we observe a discernible emphasis on the importance of `<h1></h1>` tokens. Moreover, the model concentrates more on the pertinent facts while attenuating attention toward others. During inference, selected utterances act as evidence for the final generation, making the model more interpretable. Figure 4.9 portrays input-output qualitative case studies that elucidate the advantages of our models.

4.3.6.3 Efficiency

Efficiency takes the spotlight in Figure 4.10, proving the practical usability of the proposed method. In contrast to plain models lacking an annotator, the training process for our DW_{JL} and DW_{E2E} classifier+summarizer models demands an additional 2 hours at most on DialogSum, while it only takes up to 16 extra minutes on SAMSum. During inference, ROBERTA-large runs in ≈ 8 seconds for the entire DialogSum test set and ≈ 13 seconds for the SAMSum test set. As a result, the additional cost introduced by the annotator is negligible and cannot exceed the runtime variability effect between different runs. In fact, the overall inference times of DW_{JL} and DW_{E2E} are even shorter than those of BART. The shorter average length of the summaries produced by our models adds to the rationale for this time efficiency (Table 4.8).

4.3.6.4 Human Evaluation

In the quest to better gauge summarization merits, we conduct an in-depth human evaluation of three highly-comparable models on SAMSum: the previous state-of-the-art holder MV-BART and ours BART(DW_{JL}), BART(DW_{EE}). The exclusive focus on SAMSum is firmly supported by its widespread popularity and frequent adoption as the sole benchmark in previous research work [Liu et al., 2021, Liu and Chen, 2021, Chen and Yang, 2020]. Motivated by [Narayan et al., 2018, Fabbri et al., 2019, Huang et al., 2023b], we use a direct comparison strategy, which has been shown to be more reliable, sensitive, and less labor intensive than rating scales. We sample 50 instances from the test set. For each instance, 3 English-proficient graders are presented with summaries inferred from 2 out of 3 sources and asked to select the better one with respect to 3 dimensions. A "Tie" is declared if a judge perceives the two summaries to be of equal quality. When all possible pair combinations are scrutinized, the total number of preference labels per annotator is 450. 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 the target content units. *Precision* checks if the generated summary covers only the target content

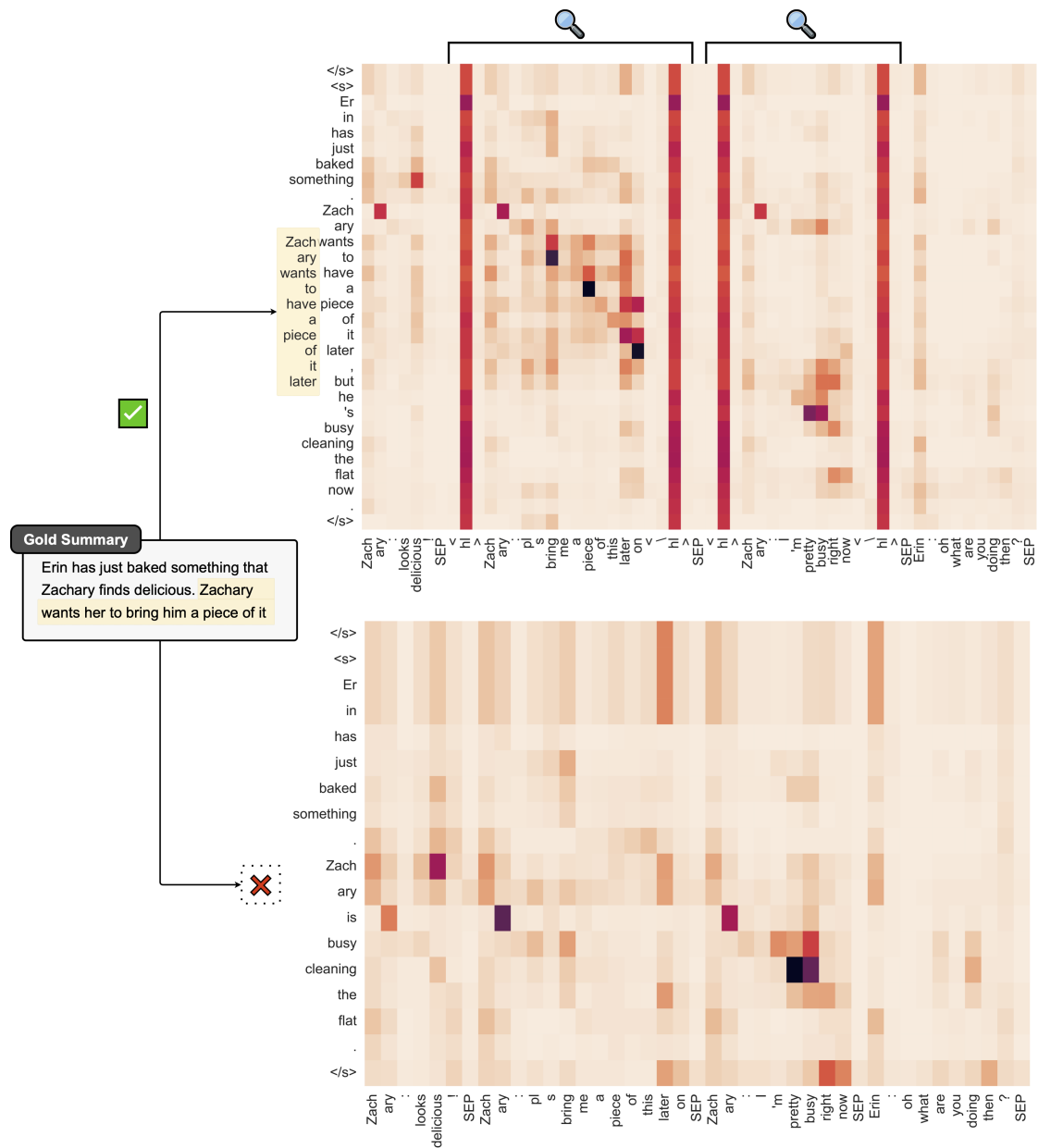


FIGURE 4.8. Attention matrix visualization, the more intense the color, the higher the weight. A comparison between BART(DW_{JL}) (left) and BART (right). The vertical axis is the generated summary, and the horizontal axis is the source dialogue. Presented tokens are partial for ease of perception. Our framework guides the model to emphasize tokens that are more likely to be relevant for the summary.

units (i.e., no superfluous or redundant information). *Faithfulness* examines whether the generated summary is factually consistent with the dialogue. The final score of each model is given by the percentage of times that its summaries are selected as the better ones, minus the percentage of times that they are not. Appendix B illustrates our setup

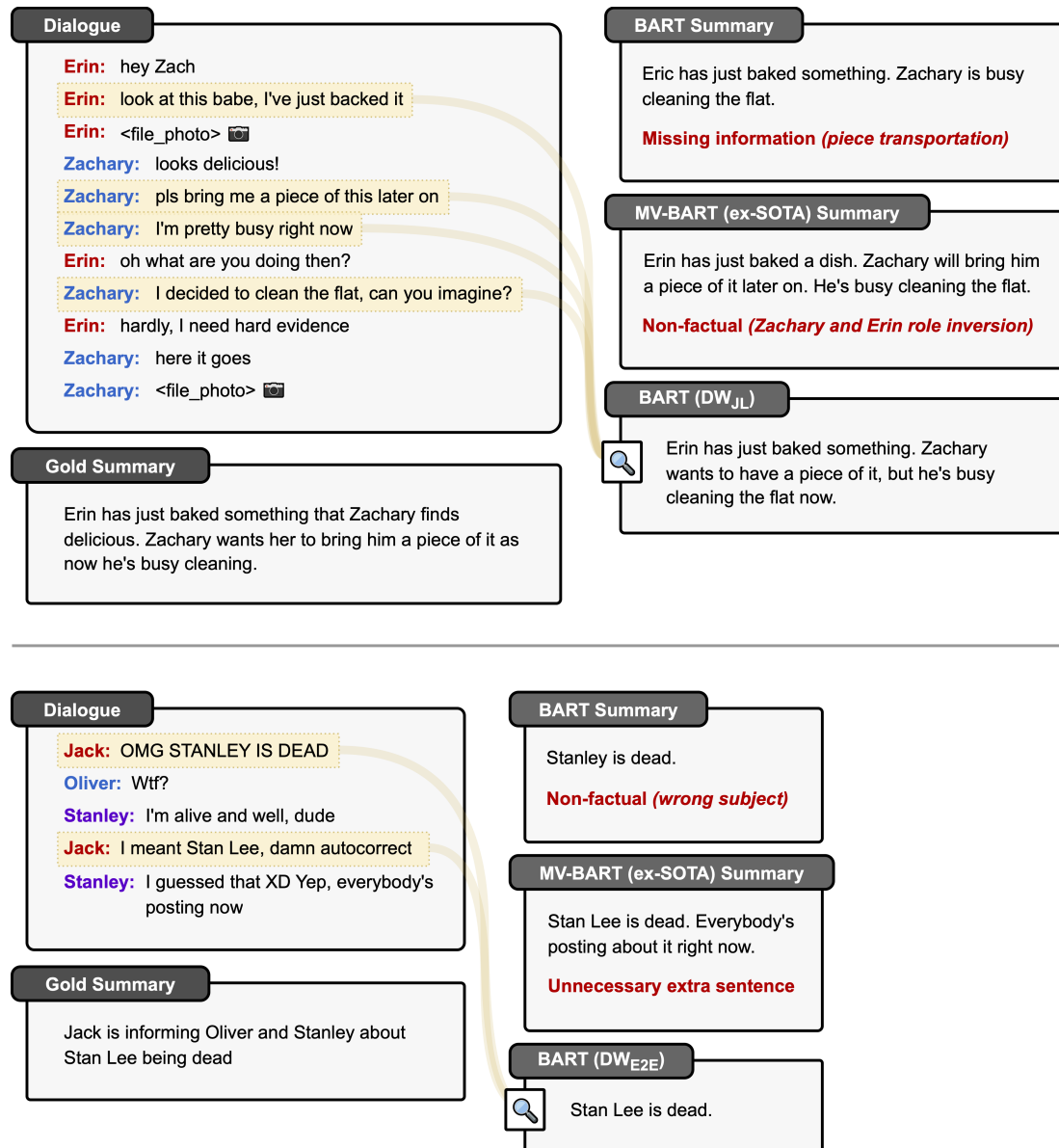


FIGURE 4.9. Qualitative examples of predicted relevant utterances (highlighted in yellow) and their assistance to high-quality summarization and interpretability. Red text indicates generation errors. Taking BART as the backbone, we illustrate the inference of our fine-tuned joint learning model (left) and end-to-end model (right).

with human instructions.

The results are showcased in Figure 4.11. Sampled instances and human judges are published for transparency and further applications.¹² The average Kendall coefficient

¹²Annotations will be released in case of acceptance.

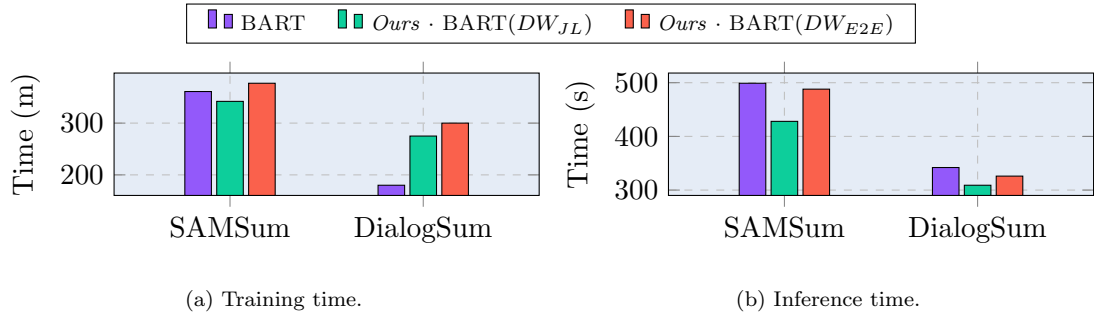


FIGURE 4.10. Runtime analysis (3-runs average) over the SAMSum and DialogSum test sets, BART-based backbone. Our model results include both annotation and generation times.

among all inter-annotator agreements is 45.25%. The annotation process took approximately 6 hours per judge, 18 hours in total. DEARWATSON models rank better on recall (+200% joint, +100% end-to-end) and faithfulness (+2324% joint, +103% end-to-end) at the cost of a moderate drop in prediction (-83% joint, -67% end-to-end). These findings reflect that the summaries inferred with our framework cover more semantic facts presented in the ground truth, successfully addressing the missing information issue. Conversely, they suggest a moderate correlation between automatic metrics and desired output properties, confirmed in subsection 4.3.6.6.

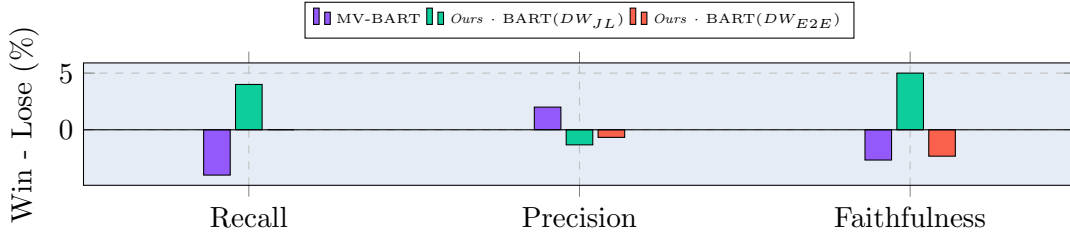


FIGURE 4.11. Human evaluation results. DEARWATSON joint learning and end-to-end learning achieve significantly higher recall and faithfulness than the previous state-of-the-art (student t-test, $p < 0.05$), corroborating the benefits of supervised input augmentation.

4.3.6.5 ChatGPT Evaluation

Scaling the model or data size has primarily raised the performance bar of NLP tasks, boosting model capacity and showing up emergent abilities [Zhao et al., 2023]. Current LLMs are general-purpose language task solvers (to some extent) and are often regarded as an initial form of artificial general intelligence [Bubeck et al., 2023]. In response to these achievements, newly published work has provided meta-evaluations of ChatGPT in zero-shot settings, attempting to ascertain whether it can evaluate text like a human expert. Gilardi et al. [Gilardi et al., 2023a] reveal that ChatGPT outperforms MTurk crowd-workers in several annotation tasks. Wang et al. [Wang et al., 2023b] substantiate

that ChatGPT surpasses previous automatic metrics in abstractive summarization and attains state-of-the-art correlation with human judgments, making it a premier NLG evaluator. In this thesis, we deploy ChatGPT to rank dialogue summarizers for the first time in the literature. In detail, we consider the same evaluation setting described in subsection 4.3.6.4 (i.e., sample, models, quality dimensions) and give aspect-specific instructions to prompt the reference-based assessment of the generated summary on a 3-point Likert scale. Figure 4.12 documents average results per model and metric. Details are reported in Appendix C.

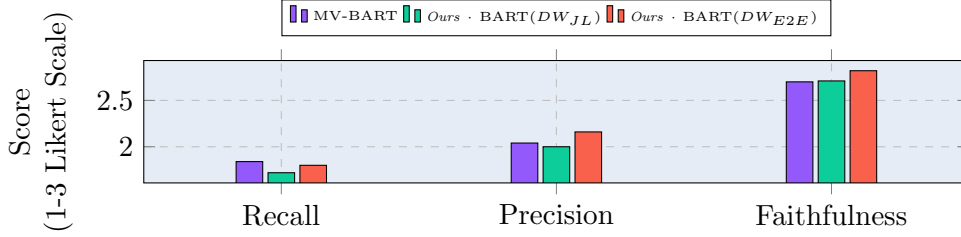


FIGURE 4.12. ChatGPT evaluation results. DEARWATSON end-to-end models exhibit higher precision and faithfulness.

4.3.6.6 Correlation with Human Judgment

We check out the correlation between automatic metrics (ChatGPT included) and human judgments in terms of semantic recall, precision, and faithfulness. We first convert human evaluation results and automatic metric scores to a scale of $\{-1, 0, 1\}$, corresponding to $\{\text{LOSE}, \text{TIE}, \text{WIN}\}$. Figure 4.13 depicts the results with Kendall’s Tau [Kendall, 1938] as the correlation measure, which evaluates the ordinal association between two quantities. Note that Kendall-Tau ranges in $[-1, 1]$, with 1 denoting a perfect positive association. Remarkably, ChatGPT correlates the most with humans, representing the most suitable technique to measure the degree to which a model resolves missing information. On the contrary, BERTScore and BARTScore perform surprisingly poorly. We hypothesize that this is due to their model-based nature and pretraining on documents different from conversations. We find ROUGE-L recall being the best metric for assessing factuality. Despite the higher proficiency of ChatGPT in capturing quality dimensions, its correlation with human raters reaches a maximum of 0.2. This indicates that ChatGPT, although a valuable metric, cannot completely substitute human judgment in the abstractive dialogue summarization field. In general terms, all metrics struggle to measure *semantic* fact recall, deviating noticeably from human labels.

4.3.7 Conclusion

In this thesis, we introduce DEARWATSON, a new annotate-then-generate framework for abstractive dialogue summarization. Drawing inspiration from the human distillation process, we train a model to predict relevant utterances and consciously build a resume

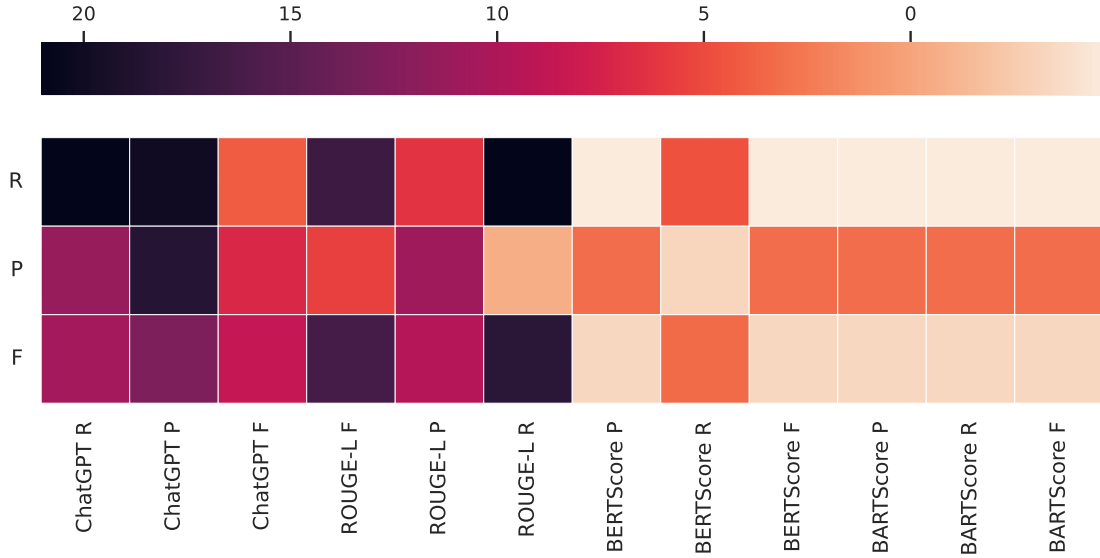


FIGURE 4.13. Kendall-Tau correlation (%) of automatic metrics with human judgments.

with respect to the essential facts. To achieve this, we jointly optimize the placement of relevance-marker tokens in the input and the summarization of the augmented dialogue. Experiments and ablation studies on SAMSum and DialogSum demonstrate that our models set new state-of-the-art results, significantly improving semantic recall and faithfulness. Cross-attention analysis unveils that our framework instructs pretrained language models to better select and preserve summary-worth content. At inference time, classified utterances also provide evidence for the output, favoring interpretability. To provide insights for future research, we measure the correlation between reported automatic metrics, prompt-guided ChatGPT scores, and human judgments.

4.3.8 Reproducibility

4.3.8.1 Implementation and Training Details

Hardware Setup and Environment Each experiment is performed on a workstation using a single Nvidia GeForce RTX3090 GPU with 24GB of dedicated memory, 64GB of RAM, and an Intel® Core™ i9-10900X1080 CPU @ 3.70GHz. The reference operating system is Ubuntu 20.04.3 LTS. To elevate portability and consistency, our development environment is constructed on top of a docker container with a de-facto standard HuggingFace image.¹³

Dataset Preprocessing We download SamSum from HuggingFace Datasets.¹⁴ We retain the original text content of conversations, encompassing cased words, slang words,

¹³<https://hub.docker.com/r/huggingface/transformers-pytorch-gpu>

¹⁴<https://huggingface.co/datasets/samsum>

typos, and emoticons. The ":" token is included in the speaker span $\mathbf{p}_i$. We replace newlines with [SEP] utterance segmenters.

Models Table 4.9 enumerates the models used in this study, linking to specific versions. They are transformer-based, have a subword vocabulary, and are pre-trained on massive corpora through denoising self-supervised tasks, i.e., reconstruction of artificially masked or corrupted spans. Given the contained size of SAMSum dialogues, no model has a maximum input size that requires truncation. It is worth mentioning that early experiments were also made with DIALOGLED [Zhong et al., 2022a] as backbone—an efficient model pre-trained for dialogue understanding and summarization on long meeting and TV series transcripts. However, the different nature and domain of pretraining dialogues ended up being unsuitable for the objectives of this thesis.

Model	# Params	Architecture			URL
		E	D	Details	
DEBERTA-V2-xlarge	900M	✓		24-layers, 1536-hidden, 24-heads	https://huggingface.co/microsoft/deberta-v2-xlarge
ROBERTA-large	355M	✓		24-layers, 1024-hidden, 16-heads	https://huggingface.co/roberta-large
GPT-2-XL	1.5B		✓	48-layers, 1600-hidden, 25-heads	https://huggingface.co/gpt2-xl
OPT-1.3B	1.3B		✓	24-layers, 2048-hidden, 32-heads	https://huggingface.co/facebook/opt-1.3b
BART-large	406M	✓	✓	12-layers, 1024-hidden, 16-heads	https://huggingface.co/facebook/bart-large
FLAN-T5-XXL	11B	✓	✓	24-layers, 4096-hidden, 64-heads	https://huggingface.co/google/flan-t5-xxl
SWING (BART-large)	406M	✓	✓	12-layers, 1024-hidden, 16-heads	https://github.com/amazon-science/AWS-SWING
MV-BART-large	~406M	✓	✓	12-layers, 1024-hidden, 16-heads	https://github.com/SALT-NLP/Multi-View-Seq2Seq
COREF-ATTN (BART-large)	~406M	✓	✓	12-layers, 1024-hidden, 16-heads	https://github.com/seq-to-mind/coref_dial_summ
DIALSENT (BART-large)	~406M	✓	✓	12-layers, 1024-hidden, 16-heads	https://github.com/jiaqisjtu/dialsent-pgg

TABLE 4.9. List of the classification (top) and summarization (bottom) models used in this study.

Baselines For each summarization baseline, we rest on the official SAMSum-specific checkpoints released by the authors, which we employ to re-run inferences and calculate metric scores. The only exception is COREF-ATTN, for which the model has not been released, and we refer directly to the official predictions.

Fine-tuning Our code is founded on Python 3.10.8, PyTorch 1.12.0 [Paszke et al., 2019], and HuggingFace Transformers [Wolf et al., 2020]. We train each model for five epochs and select the best checkpoints in the validation set with $\mathcal{R}$ score. We choose the AdamW optimizer [Loshchilov and Hutter] and leave 42 as the default global seed. Sticking to Jang et al. [Jang et al., 2017a], we anneal the Gumbel softmax value τ according to a high-low schedule. Both BART and FLAN-T5 are trained with teacher forcing: at training time, the inputs are previous tokens from the ground truth; at test time, the inputs are prior tokens predicted by the decoder. To make fine-tuning of GPT-2, OPT, and FLAN-T5 possible with our hardware, we exploit PEFT.¹⁵ Precisely, we carry out 8-bit model quantization and adopt Low-Rank Adaption (LoRA) [Hu et al., 2022] to only learn a small number of extra model parameters. Training our best model, FLAN-T5(*DW_{JL}*), requires 20GB VRAM and 25 hours; 2.13 kg CO₂e carbon footprint, 16.57 kWh energy needed.

¹⁵<https://github.com/huggingface/peft>

Hyperparameter	Search space
Top-p annotation [†]	{0.30, 0.40, 0.41, ..., 0.45*(DW_{E2E}), ..., 0.47*(DW_{JL}), ..., 0.50}
Dropout rate	0.1
Learning rate	{1e-5, 2e-5*, 3e-5, 4e-5}, linear scheduler
Optimizer	0.9 β_1 , 0.999 β_2 , 1e-2 weight decay
Batch size	2
Epochs	5 (validation every epoch)
τ [†]	{1, ..., 14*, ..., 21}
Lora_dropout [‡]	0.1
Lora_rank [‡]	{16*, 32, 64}
Lora_alpha [‡]	{32*, 64, 128}
Decoding strategy	beam search, n_beams=5 min_length=8, max_length=100 repetition_penalty=1

TABLE 4.10. Explored hyperparameters along with their empirical search grid. Training time (top) and inference time (bottom). * marks the final picked values. † and ‡ are specific for end-to-end strategies and large language models, respectively.

Hyperparameters We list the hyperparameters used to train our DEARWATSON models in Table 4.13. Surveyed values include default settings [Lewis et al., 2020b, Chung et al., 2022b]; final choices result from a grid search.

Number of Parameters The architectural sizes of our solutions equal the sum of the classifier and summarizer parameters, independently of the training strategy. By combining a ROBERTA-large annotator and a BART-large generator, our BART($DW_{JL/E2E}$) model is $\sim 761.6\text{M}$ (all trained). Similarly, FLAN-T5($DW_{JL/E2E}$) is $\sim 3.2\text{B}$ (of which only 363.8M are trained).

Experiment Tracking We track all our trainings with Weights & Biases¹⁶ and monitor CO2 emissions with CodeCarbon.¹⁷

4.3.8.2 Metrics

We quantify automatic metric scores using NLG-METRICVERSE [Frisoni et al., 2022]. Note that different libraries may result in different scores. Table 4.11 lists all hyperparameters. Owing to the grander correlation with human judgment, we compute BERTScore with DEBERTA-xlarge instead of the default ROBERTA-large, as recommended by the authors from version 0.3.11. To increase interpretability and avoid slight range variations, we set `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 $\mathbf{x}$,

¹⁶<https://wandb.ai>

¹⁷<https://github.com/mlco2/codecarbon>

Metric	Definition	Bound	Hyperparameters
ROUGE	Unigrams, bigrams, and longest common subsequence lexical overlaps	$[0, 1] \uparrow$	<code>rouge_types=["rouge1", "rouge2", "rougeL"],</code> <code>use_aggregator=True,</code> <code>use_stemmer=True,</code> <code>metric_to_select="fmeasure"</code> <code>model_type="microsoft/deberta-xlarge-mnli",</code> <code>idf=True, batch_size=64, nthreads=4,</code> <code>rescale_with_baseline=True,*</code> <code>use_fast_tokenizer=False,</code> <code>return_average_scores=False</code>
BERTScore	IDF-weighted n-gram hard-alignment via contextualized embeddings	$[-1, 1]^* \uparrow$	<code>model_checkpoint="bartscore-large-cnn",</code> <code>batch_size=4, segment_scores=False</code>
BARTScore	Semantic multi-perspective evaluation as the logarithmic probability of generating a text conditioned on another one	$]-\infty, 0[\uparrow$	<code>model_checkpoint="bartscore-large-cnn",</code> <code>batch_size=4, segment_scores=False</code>

TABLE 4.11. Hyperparameters initialization for utilized NLG metrics. Arrows indicate the reading key (i.e., $\uparrow$ = higher is better).

where θ are the weights of a BART model. Because of this generative approach, the evaluation dimensions vary depending on how $\mathbf{y}$ and $\mathbf{x}$ are defined. As for the other metrics, we consider the Recall, Precision, and F1 settings, thereby feeding BARTScore with a summary hypothesis ($\mathbf{h}$) and a summary reference ($\mathbf{r}$). Recall ($\mathbf{h} \rightarrow \mathbf{r}, p(\mathbf{r}|\mathbf{h}, \theta)$) quantifies how easily a gold reference could be generated by the hypothesis (i.e., semantic coverage). Precision ($\mathbf{r} \rightarrow \mathbf{h}, p(\mathbf{h}|\mathbf{r}, \theta)$) assesses how likely the summary hypothesis could be constructed based on the gold reference. F score ($\mathbf{h} \leftrightarrow \mathbf{r}$) takes the arithmetic average of recall and precision.

4.3.9 ChatGPT Evaluation Details

We leverage the ChatGPT API and treat the `gpt-3.5-turbo` model as a reference-based metric for judging artificial summaries. We feed predictions individually, utilizing the prompt in Figure 4.14—which, after several attempts, results as the best one. The prompt makes the model aware of task and aspect details, forcing ChatGPT to output scores only.

- *Recall*. 1 = "reference contents are not covered at all". 3 = "reference contents are fully covered".
- *Precision*. 1 = "the generated summary has many contents not covered by the reference (unnecessary or redundant information)". 3 = "the generated summary contains only reference contents".
- *Faithfulness*. 1 = "the generated summary is not semantically consistent with the dialogue (hallucinations, polarity inversions, entity misuse)". 3 = "the generated summary is perfectly factual with respect to the dialogue".

A new chat session is created for every summary to ensure that ChatGPT results (i.e., single digits) are not influenced by the annotation history. As for decoding hyperparameters, we leave default values (e.g., temperature=1, top-p=1).

Template You are an expert English proficient annotator working on abstractive dialogue summarization. Score the following generated summary given the corresponding [input] with respect to [metric] on a scale from 1 to 3, where a score of 1 means [definition] and a score of 3 means [definition]. Dialogue Summary: [text] Summary: [text]	Example You are an expert English proficient annotator working on abstractive dialogue summarization. Score the following generated summary given the corresponding reference with respect to recall on a scale from 1 to 3, where a score of 1 means "reference contents are not covered at all" and a score of 3 means "reference contents are fully covered". Dialogue Summary: Jack is informing Oliver and Stanley about Stan Lee being dead Summary: Stan Lee is dead
--	--

FIGURE 4.14. ChatGPT annotation prompt (target: value on a 3-point Likert scale). [metric] equals "recall", "precision", or "faithfulness". The [input] can be a "resume" (recall/precision) or a "document" (faithfulness). [definition] spans explain the meaning of boundary min/max scores for the metric under investigation.

4.3.10 Human Evaluation Details

The interface with human evaluation instructions is sketched in Figure 4.15.

Instructions for Human Evaluators					
Two neural models attempt to summarize a dialogue while retaining the salient information. Which summary do you think is better in the following three dimensions? • Recall: Does the generated summary cover <i>all</i> the reference summary contents? • Precision: Does the generated summary cover <i>only</i> the reference summary contents (i.e., no superfluous or redundant information)? • Faithfulness: Is the summary semantically consistent w.r.t. the original dialogue?					
Dialogue [...]		Reference Summary [...]			
Read carefully the following two summaries and then select the radio buttons below about the summary that you prefer					
Summary 1 [...]		Summary 2 [...]			
Which is...					
1	Higher Recall?	2	Higher Precision?	3	Higher Faithfulness?
	☐ ☐		☐ ☐		☐ ☐

FIGURE 4.15. Human assessment interface.

4.3.11 Scientific Artifacts

The licenses for all the models and software used in this thesis are listed below in parentheses: NLTK (Apache License 2.0), py-ROUGE (Apache License 2.0), BERTScore (MIT License) BARTScore (Apache License 2.0), SAMSum (CC BY-NC-ND 4.0), GPT-2 (MIT License), OPT (non-commercial), DeBERTa (MIT License), RoBERTa (GPL-2.0 License), MV-BART (MIT License), Coref-ATTN (not specified), S-BART (MIT License), BART-DialoGPTAnn (not specified), SWING (Apache License 2.0), BART (MIT License), FLAN-T5 (Apache 2.0), DialSent (not specified).

4.4

MEMEWEAVER: Inter-Meme Graph Reasoning for Sexism and Misogyny Detection

Originally published as: Paolo Italiani, David Gimeno-Gomez, Luca Ragazzi, Gianluca Moro, and Paolo Rosso. *Memeweaver: Inter-meme graph reasoning for sexism and misogyny detection*, 2026a. URL <https://arxiv.org/abs/2601.08684>

Abstract

Women are twice as likely as men to face online harassment due to their gender. Despite recent advances in multimodal content moderation, most approaches still overlook the social dynamics behind this phenomenon, where perpetrators reinforce prejudices and group identity within like-minded communities. Graph-based methods offer a promising way to capture such interactions, yet existing solutions remain limited by heuristic graph construction, shallow modality fusion, and instance-level reasoning. In this work, we present MEMEWEAVER, an end-to-end trainable multimodal framework for detecting sexism and misogyny through a novel inter-meme graph reasoning mechanism. We systematically evaluate multiple visual-textual fusion strategies and show that our approach consistently outperforms state-of-the-art baselines on the MAMI and EXIST benchmarks, while achieving faster training convergence. Further analyses reveal that the learned graph structure captures semantically meaningful patterns, offering valuable insights into the relational nature of online hate.

4.4.1 Introduction

Hate speech on social media is rarely an isolated act: it spreads through interactions among users. Psychological research [Walther, 2022] shows that perpetrators often gain social approval by sharing content with like-minded individuals, reinforcing prejudices and group identity. These dynamics, compounded with user anonymity [Kowalski and Whittaker, 2015], amplify harmful narratives and foster in-group codes and jokes, making content subtler, context-dependent, and harder to detect.

Within the spectrum of hate speech [Herz and Molnár, 2012], women remain one of the most disproportionately targeted groups. They are twice as likely as men to be harassed online due to their gender [Duggan, 2017], often within communities where such hateful discourses are normalized and perpetuated [Fontanella et al., 2024]. Such abuse manifests primarily as *sexism*—prejudice or discrimination often rooted in stereotypes and socially accepted forms of bias [Glick and Fiske, 2001]—and *misogyny*, marked by deeper hostility, contempt, and explicit aggression [Richardson-Self, 2018]. Persistent exposure to both has been shown to harm women’s self-esteem, restrict career ambitions, and reinforce traditional gender roles [Bradley-Geist et al., 2015]. The rapid dissemination

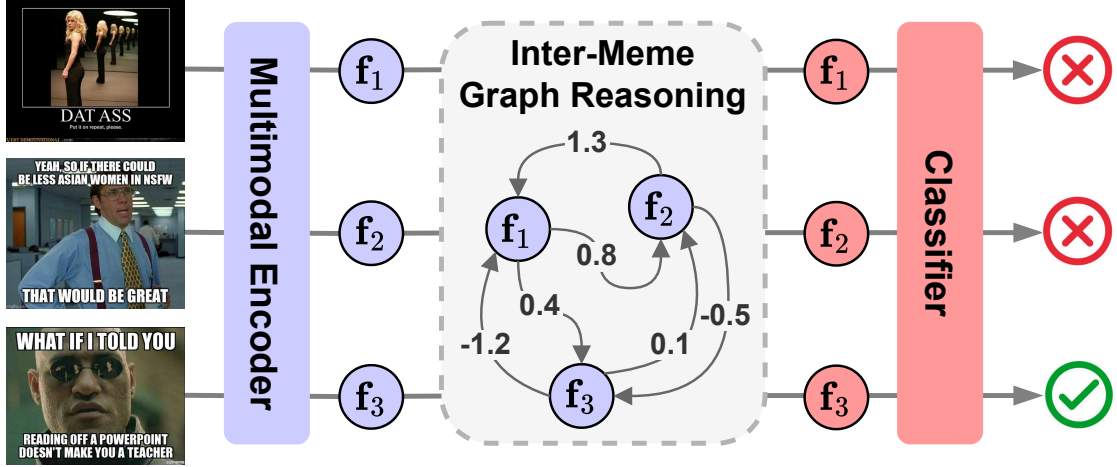


FIGURE 4.16. Overview of MemeWeaver, our graph-based framework for end-to-end sexism/misogyny detection in memes. Motivated by the social dynamics of online hate, it models batches of memes (3 shown) as inter-connected nodes for more effective learning.

of such content in online communities makes effective regulation challenging [Herz and Molnár, 2012], motivating the development of automated detection methods [Luo et al., 2025].

Despite recent progress, most computational approaches overlook the social dynamics underlying online hate. Graph-based models [Zhang et al., 2019, Rehman et al., 2025, Xu et al., 2025] provide a promising avenue by capturing inter-sample relationships. However, existing solutions still rely on heuristic preprocessing, shallow modality fusion, and instance-level reasoning, which limit generalization and adaptability across domains.

In this thesis, we introduce MEMEWEAVER (see Figure 4.16), an end-to-end trainable framework for detecting sexism and misogyny in memes. Our approach integrates three key ideas. *First*, memes are inherently multimodal and context-dependent: they combine text embedded in images with visual scenes, and their meaning often hinges on humor, irony, or cultural references that may obscure harmful intent [Hodson et al., 2010, Drucker et al., 2014, Plaza et al., 2024, Cocchieri et al., 2025]. To address this, we verbalize the text content of each meme and optionally enrich it with captions generated by a multimodal LLM, capturing both surface descriptions and higher-level interpretations. *Second*, we encode text and images with CLIP-based encoders and explore alternative fusion strategies that move beyond simple concatenation, enabling richer multimodal representations. *Third*, and most importantly, we introduce a novel Inter-Meme Graph Reasoning (IMGR) mechanism that models relationships across memes within each training batch. This design allows the system to automatically “weave” latent affinities between memes, capturing relational patterns without relying on heuristic graph construction.

We conduct extensive experiments on two complementary meme benchmarks: MAMI, a dataset targeting misogyny in English memes, and EXIST, a multilingual corpus focused on sexism with both English and Spanish memes. We compare MEMEWEAVER

against established baselines as well as recent large multimodal language models (LLMs). Our main contributions are threefold:

- **State-of-the-art results.** MEMEWEAVER consistently outperforms strong baselines on both datasets, improving over CLIP fine-tuning by ≈ 5 points on average while converging faster during training.
- **Fusion analysis.** We evaluate alternative text-image fusion mechanisms, showing that their effectiveness depends on dataset characteristics and that higher model complexity does not necessarily yield better performance.
- **Graph insights.** We analyze the learned embeddings and inter-meme graph structure, revealing semantically meaningful affinity patterns that offer insights into the relational dynamics of online hate.

4.4.2 Related Work

Long-standing research has addressed the detection of sexism and misogyny in online content. This subsection outlines the shift from early text-only methods to recent multimodal approaches, highlighting open challenges in graph-based reasoning.

Sexism and Misogyny Detection Early work relied on statistical text features (e.g., n-grams, TF-IDF) with classical machine learning classifiers [Jha and Mamidi, 2017, Anzovino et al., 2018]. Deep learning, and in particular pre-trained transformers such as BERT, improved contextual understanding [Parikh et al., 2021, de Paula et al., 2023]. More recently, prompt-based techniques with LLMs have been explored for nuanced comprehension [Tian et al., 2023, Samani et al., 2025]. Given the multimodal nature of social media posts [Luo et al., 2025], research has shifted to text-vision integration [Gomez et al., 2020], especially in memes [Fersini et al., 2022, Plaza et al., 2025]. Captioning-based methods [Rizzi et al., 2023, Plaza et al., 2024], cross-modal contrastive learning [Rizzi et al., 2024], multimodal LLMs [Cao et al., 2022, Xu et al., 2025], and prompt-based visual QA approaches such as Pro-Cap [Cao et al., 2023] have been widely adopted. More recent studies extend to in-the-wild videos, incorporating voice and visual cues [Arcos and Rosso, 2024, Plaza et al., 2025]. Despite these advances, systems still struggle with implicit, context-dependent sexism and with biases in data and models, pointing to the need for richer reasoning mechanisms such as graph-based approaches.

Multimodal Graph Modeling Graph-based models, particularly GCNs [Zhang et al., 2019], are increasingly used to capture structural and semantic relations across modalities, with applications in tasks such as emotion recognition [Zhu et al., 2022] and scene-text retrieval [Maffa et al., 2021, Li et al., 2024b]. In online hate speech detection, they have also shown promise. Xu et al. [2025] combined LLMs with hypergraph learning to integrate text and visual features and capture implicit semantics, while Rehman et al. [2025] enriched unimodal representations through gated cross-attention rather than simple concatenation. However, both approaches relied on similarity-based heuristics to

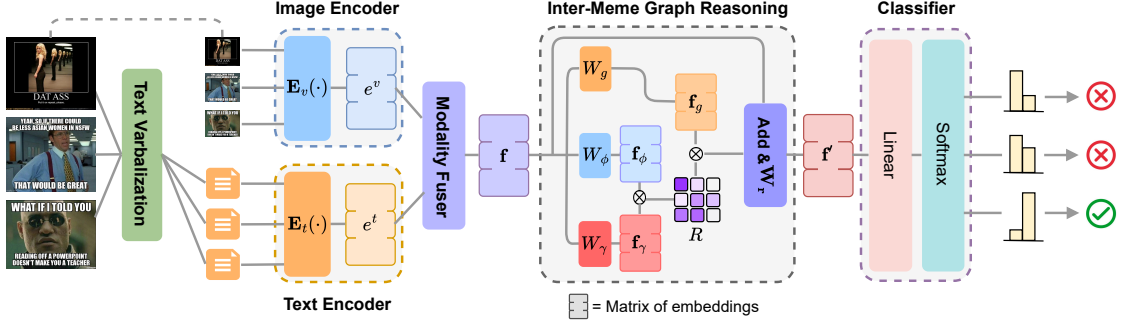


FIGURE 4.17. **Architecture of MemeWeaver.** Each meme undergoes text extraction (OCR, optionally enriched with LLM-generated captions) and is encoded by separate text and image encoders. The embeddings are fused, refined via inter-meme graph reasoning within each batch, and classified. Example shown with batch size $m = 3$.

define the graph. Only Hebert et al. [2024] introduced learnable components, though their attention-based method still depends on heuristics such as the number of user mentions in social media posts.

Research Gaps Despite recent progress, two challenges remain central. First, most multimodal hate speech detection methods—including those targeting sexism and misogyny—rely on shallow fusion strategies, often limited to feature concatenation. Second, existing graph-based approaches typically construct structures through heuristic rules, restricting adaptability and limiting their ability to capture inter-sample relations. These limitations call for fully learnable frameworks that can jointly model multimodal content and social interactions in an end-to-end manner. Our work explores whether such a design can uncover generalizable relational patterns or whether optimal configurations are inherently domain-specific.

4.4.3 MemeWeaver

We introduce MEMEWEAVER, a framework for detecting sexist and misogynistic memes, which “weaves” together textual and visual features while modeling relations among memes at the batch level.

4.4.3.1 Problem Statement

Let $\mathcal{X} = \{x_1, \dots, x_n\}$ be a multimodal dataset of memes, processed in mini-batches of size m . Each instance $x_j = (t_j, v_j)$ consists of a textual component t_j (OCR-extracted text, optionally enriched with an LLM-generated caption) and an image v_j , annotated with a binary label $y_j \in \{0, 1\}$, where 1 denotes a sexist/misogynistic meme and 0 otherwise. The goal is to train a multimodal classifier f that, given a novel meme (t, v) , predicts its label y .

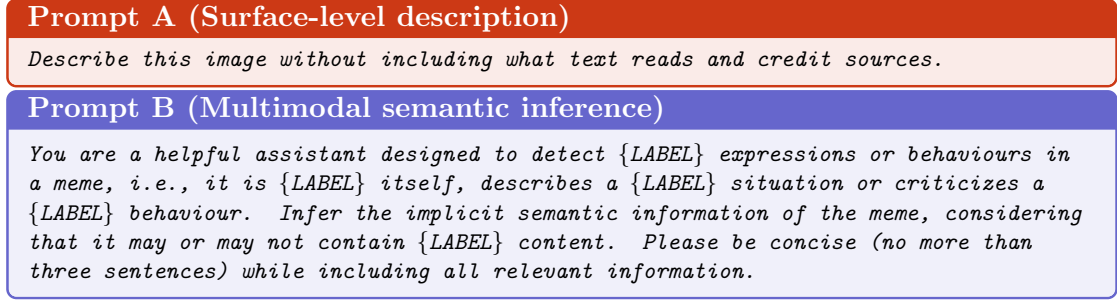


FIGURE 4.18. **Prompts used for LLM-based meme captioning.** The LABEL placeholder was set to *misogynistic* or *sexist* depending on the downstream dataset.

4.4.3.2 Model Architecture

Figure 4.17 shows the architecture of MEMEWEAVER. Each meme is an image that includes both a visual scene and overlaid text. Our pipeline begins with a verbalization step, which extracts the text embedded in the image.¹⁸ This content may be optionally enriched with an LLM-generated caption. The extracted text is then encoded by a text encoder, while the complete meme image is processed by a separate image encoder. The two encoders yield modality-specific representations, which are subsequently fused into a multimodal embedding. To capture inter-sample relationships, we weave connections among memes within each training batch, modeling the batch as a fully connected graph. Finally, the context-enriched embeddings are fed into a classifier, with all modules trained end-to-end.

LLM-based Captioning Inspired by Xu et al. [2025], we leverage a multimodal LLM to generate not only surface-level descriptions but also high-level interpretations of memes. Specifically, we used Qwen2.5-VL-7B-Instruct¹⁹ and designed two prompt strategies (see Figure 4.18). The generated captions are concatenated with the OCR-extracted meme text, separated by the special token [CPT]; we evaluate both OCR-only and OCR+caption variants. Examples are provided in the Appendix.

Text Encoder Given a batch of memes, we process their textual components $\{t_1, \dots, t_m\}$ with a transformer-based text encoder $\mathbf{E}_t(\cdot)$, yielding hidden representations:

$$\begin{aligned} e_j^t &= \mathbf{E}_t(t_j), \quad j = 1, \dots, m \\ e^t &= [e_1^t, \dots, e_m^t] \in \mathbb{R}^{m \times \frac{d}{2}} \end{aligned} \tag{4.22}$$

where $d = 1536$ denotes the overall multimodal hidden dimensionality, so that each encoder outputs vectors of size $d/2$.

¹⁸In the datasets used for our experiments, this text annotation is already provided by the dataset creators.

¹⁹Run with default precision and decoding settings from the official configuration file.

Image Encoder Similarly, given the same batch of memes, we process the meme images $\{v_1, \dots, v_m\}$ with an image encoder $\mathbf{E}_v(\cdot)$, yielding hidden representations:

$$\begin{aligned} e_j^v &= \mathbf{E}_v(v_j), \quad j = 1, \dots, m \\ e^v &= [e_1^v, \dots, e_m^v] \in \mathbb{R}^{m \times \frac{d}{2}} \end{aligned} \quad (4.23)$$

with the same dimensionality convention as in the text encoder. Both the text and image encoders are initialized from CLIP and fine-tuned jointly with the rest of the architecture.

Modality Fuser In addition to simple concatenation, we consider two fusion mechanisms to combine the text and image embeddings into unified multimodal representations $\mathbf{f} = [f_1, \dots, f_m]$.

(1) *Multi-modal Factorized Bilinear (MFB) pooling* [Kim et al., 2017] projects each modality into a shared space and applies element-wise interaction:

$$f_j = (e_j^t U^\top) \circ (e_j^v V^\top), \quad j = 1, \dots, m \quad (4.24)$$

where $U, V \in \mathbb{R}^{\frac{d}{2} \times d}$ are trainable projection matrices and $\circ$ denotes the Hadamard product.

(2) *Gated Multimodal Unit (GMU)* [Ovalle et al., 2017] learns a gating vector to adaptively control each modality’s contribution:

$$\begin{aligned} z_j &= \sigma([e_j^t, e_j^v] U_z), \\ f_j &= z_j \odot \tanh(e_j^v U_v^\top) \\ &\quad + (1 - z_j) \odot \tanh(e_j^t U_t^\top), \quad j = 1, \dots, m \end{aligned} \quad (4.25)$$

where $U_t, U_v \in \mathbb{R}^{\frac{d}{2} \times d}$ and $U_z \in \mathbb{R}^{d \times d}$ are trainable matrices, σ is the sigmoid, $[\cdot, \cdot]$ denotes concatenation, and $\odot$ denotes multiplication.

Inter-Meme Graph Reasoning (IMGR) Inspired by prior work on relational reasoning [Li et al., 2019a, Mafra et al., 2021], we enhance multimodal representations by modeling relationships *across* memes in a batch, rather than within a single instance (e.g., linking image regions or textual tokens). We compute an affinity matrix $R \in \mathbb{R}^{m \times m}$ capturing pairwise meme similarities:

$$R = (\mathbf{f} W_\phi)(\mathbf{f} W_\gamma)^\top \quad (4.26)$$

where $\mathbf{f} = [f_1, \dots, f_m] \in \mathbb{R}^{m \times d}$ are fused embeddings and $W_\phi, W_\gamma \in \mathbb{R}^{d \times d}$ are trainable projections. This yields a fully connected graph $G = (\mathbf{f}, R)$, with nodes f_i (memes) and edge weights R_{ij} (their relations). Following Li et al. [2019a], we then apply a graph-based message passing layer with residual connections:

$$\mathbf{f}' = (\hat{R} \mathbf{f} W_g) W_r + \mathbf{f} \quad (4.27)$$

Dataset	No. of Samples (% Misogyny/Sexism)		
	<i>Training</i>	<i>Validation</i>	<i>Test</i>
MAMI	9,000 (50.0%)	1,000 (50.0%)	1,000 (50.0%)
EXIST	3,235 (65.7%)	404 (67.3%)	405 (65.7%)

TABLE 4.12. **Dataset statistics.** We highlight the percentage of sexist and misogynistic memes in each split.

where $W_r, W_g \in \mathbb{R}^{d \times d}$ are trainable projections and $\hat{R} = \frac{R}{m}$ is the uniform rescaled affinity matrix. The final representations $\mathbf{f}' = [f'_1, \dots, f'_m]$ are enriched with batch-level context, enabling each meme to incorporate information from its peers.

Classifier and Loss Function The classifier consists of a linear projection $W_c \in \mathbb{R}^{d \times 2}$ producing logits, followed by a softmax that yields probabilities $P = [p_1, \dots, p_m] \in \mathbb{R}^{m \times 2}$, where each $p_j \in \mathbb{R}^2$ is the predicted distribution for meme j . All modules of our architecture are trained end-to-end using cross-entropy loss:

$$\mathcal{L}_{ce} = -\frac{1}{m} \sum_{j=1}^m \sum_{c \in \{0,1\}} \mathbf{1}[y_j = c] \log p_{j,c} \quad (4.28)$$

where $y_j \in \{0, 1\}$ is the gold label for instance j .

4.4.4 Experimental Setup

4.4.4.1 Datasets

To capture domain and cultural diversity, we evaluate our method on two distinct multimodal meme datasets for misogyny and sexism detection, framing the task as binary classification following prior approaches [Cao et al., 2023, Xu et al., 2025].

(1) **MAMI** [Fersini et al., 2022] is a curated SemEval dataset focused specifically on *misogyny* in memes, collected from Twitter, Reddit, and meme-focused websites. It provides fine-grained subtype annotations such as shaming, stereotyping, objectification, and incitement to violence; we assign the misogyny label if a meme belongs to any subtype. This widely used benchmark serves as our primary dataset for misogyny detection in English memes.

(2) **EXIST** [Plaza et al., 2025] is a CLEF shared task on *sexism* detection across social media, including tweets and videos. Here we focus on Task 2, targeting memes. The dataset was built from 250 sexism-related queries on Google Images, and includes subtype labels such as ideological inequality, stereotyping and dominance, objectification, and sexual violence. Since each meme was annotated by three annotators, we derived the final sexism label via majority voting. Unlike MAMI, EXIST contains memes in both English and Spanish, allowing cross-lingual and cross-cultural evaluation.

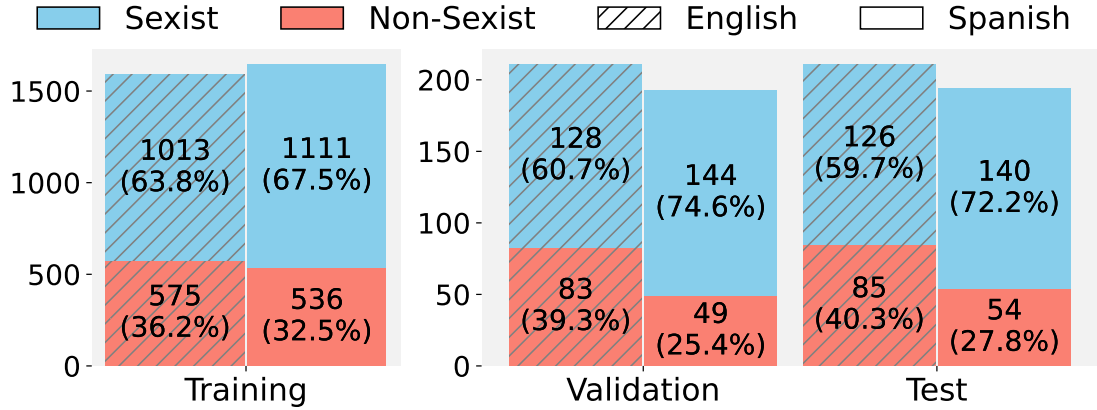


FIGURE 4.19. **Language distribution in EXIST.** Ratio of sexist vs. non-sexist memes across splits.

Table 4.12 shows the dataset distributions. MAMI is balanced (50% misogynistic memes), while EXIST is skewed toward sexist content ($\approx 66\%$). Figure 4.19 further breaks down EXIST by language and class across splits: the training set is relatively balanced, but the skew becomes more pronounced in the validation and test sets, especially for Spanish. For MAMI we rely on the official splits, whereas for EXIST we define a custom 80/10/10 train/val/test split, since no validation set is provided and the official test set is not publicly available.²⁰ Both datasets already come with OCR-extracted text.

4.4.4.2 Baselines

We use the widely adopted multimodal model CLIP [Radford et al., 2021]²¹ as the main backbone of MEMEWEAVER; results with alternative models are reported in the Appendix. We choose CLIP over larger multimodal LLMs because its lightweight design allows for larger effective batch sizes, thereby facilitating richer inter-meme relationships within IMGR. Unlike for EXIST, we use the original test set for MAMI, and thus report results against established baselines: EF-CaTrBERT [Khan and Fu, 2021], PromptHate [Cao et al., 2022], Pro-Cap [Cao et al., 2023], and HyperHatePrompt [Xu et al., 2025]. In addition, we compare against recent multimodal LLMs, such as Qwen2.5-VL [Bai et al., 2025], Phi-4 [Abdin et al., 2024], and GPT-4o-mini [Hurst et al., 2024], evaluated in a zero-shot setting using instruction-following prompts (details are provided in the Appendix).

²⁰This ensures consistent internal evaluation, though results are not directly comparable to leaderboard scores.

²¹HuggingFace checkpoint: openai/clip-vit-large-patch14

Setting	Range	MAMI	EXIST
Training batch	[10, 180]	20	64
Inference batch	[10, 180]	41	27
Class. threshold	[0.010, 0.999]	0.657	0.434
Modality fusion	N/A	MFB	Concat
Image captioning	N/A	–	Prompt A

TABLE 4.13. **Optimal settings.** Ranges denote search intervals; final values were selected on the validation sets.

Method	MAMI <i>English</i>			<i>All</i>			EXIST <i>English</i>			<i>Spanish</i>		
	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC
Phi-4	69.8	70.0	-	61.4	62.5	-	66.4	65.8	-	58.3	56.3	-
GPT-4o-mini	71.7	73.1	-	64.4	64.4	-	70.1	70.1	-	57.8	58.2	-
Qwen2.5-VL	74.4	74.8	-	71.4	<u>70.0</u>	-	72.5	<u>72.1</u>	-	70.1	66.7	-
MEMEWEAVER												
w/ IMGR	77.6*	77.4*	83.4*	73.3	67.9*	<u>76.9*</u>	71.6	68.4*	<u>77.0*</u>	75.3	66.3*	76.5*
w/o IMGR	71.9	71.8	79.4	71.1	60.5	71.2	69.2	61.6	74.8	73.2	58.3	65.9
+ Prompt A												
w/ IMGR	74.8	74.4	80.1	76.3	71.6*	75.7*	76.3*	73.6*	72.9	<u>75.3</u>	<u>67.2</u>	66.4
w/o IMGR	69.2	68.9	77.5	72.1	61.8	73.5	68.2	59.5	75.1	76.3	64.4	72.1
+ Prompt B												
w/ IMGR	<u>76.3*</u>	<u>76.6*</u>	<u>83.5*</u>	<u>73.6</u>	67.3	78.9*	<u>73.9</u>	71.3	80.4*	73.2	59.1	<u>76.4</u>
w/o IMGR	72.7	71.9	78.8	73.3	69.7	74.3	73.5	71.0	76.3	73.2	67.7	72.3

TABLE 4.14. **Comparison to state of the art on test sets.** Best results in bold, second best underlined. * denotes statistically significant improvements ($\alpha = 0.05$, Koehn, 2004). Ablation studies confirm the importance of IMGR.

	Acc	F1	AUC
Prior Work			
EF-CaTrBERT [2021]	67.8	67.2	74.9
PromptHate [2022]	71.1	70.8	80.8
Pro-Cap [2023]	73.3	72.4	83.2
HyperHatePrompt [2025]	<u>75.3</u>	<u>75.1</u>	84.3
MEMEWEAVER (Ours)	77.6	77.4	<u>83.4</u>

TABLE 4.15. **Our best MemeWeaver vs. state of the art on MAMI.** Best scores in bold, second-best underlined.

4.4.4.3 Implementation Details

Environment All experiments run on a workstation equipped with an NVIDIA GeForce RTX 3090 GPU (24 GB VRAM), 64 GB system RAM, and an Intel® Core™ i9-10900X CPU @ 3.70GHz. Our implementation builds on Python 3.10.12, PyTorch 2.2.1, and HuggingFace Transformers 4.44.0.

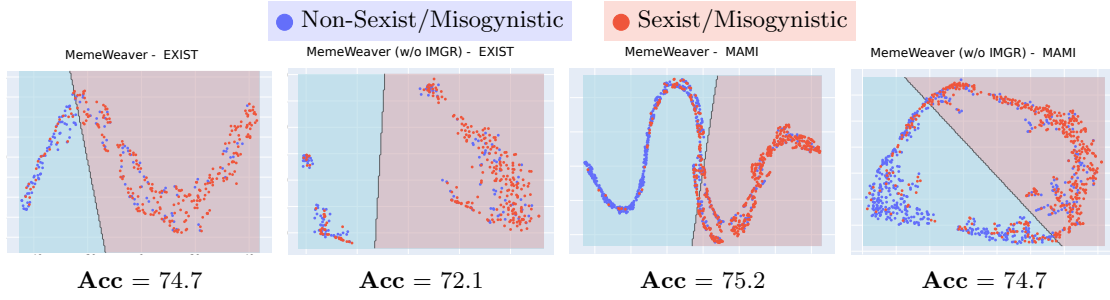


FIGURE 4.20. **Embedding space analysis.** 2D visualizations of MAMI and EXIST test set embeddings $\mathbf{f}'$ and $\mathbf{f}$, generated with and without MEMEWEAVER’s IMGR, respectively. Decision boundaries from a linear classifier show higher accuracy on $\mathbf{f}'$ (see scores below plots), underscoring the benefit of IMGR’s representations.

Multilingual Setup For EXIST, we translated the Spanish split into English using Google Translate, since CLIP is English-only. Given the short and simple nature of the texts, automatic translation is highly reliable. We manually inspected the entire translated set and found no systematic errors or meaning distortions, ensuring negligible translation bias. This choice enables a unified backbone across datasets, avoiding confounding effects from heterogeneous multilingual encoders and allowing a fair assessment of IMGR. It also represents a practical strategy for handling multilingual memes when only monolingual encoders are available.

Method Settings For the EXIST dataset, which is relatively low-resource in terms of available training data—thus also simulating a more realistic scenario with limited supervision—we initialize from the checkpoint pretrained on MAMI before fine-tuning. All models are trained for 3 epochs with the AdamW optimizer (learning rate $5e-6$) and a linear scheduler, using a fixed random seed of 42 for reproducibility. Training and inference batch sizes are tuned on the validation set, as they directly determine how many instances can interact within the IMGR module and thus critically impact MEMEWEAVER’s performance. Finally, rather than adopting the default 0.5 classification threshold, we follow Xu et al. [2025] and select the cutoff that maximizes F1 on the validation set. Table 4.13 summarizes all final hyperparameters, including not only optimization settings but also the selected fusion strategy and the use of LLM-based captioning along with the best-performing prompt.

Evaluation Metrics For each model, we retain the checkpoint with the highest macro-F1 score on the validation set. We report three complementary metrics: Accuracy (Acc), macro-F1, and Area Under the Curve (AUC). Among these, macro-F1 serves as our primary metric, as it equally weights both classes and is therefore more reliable under the label imbalance in the EXIST dataset.²²

²²For generative LLMs, which output hard labels rather than probability distributions, AUC is not reported.

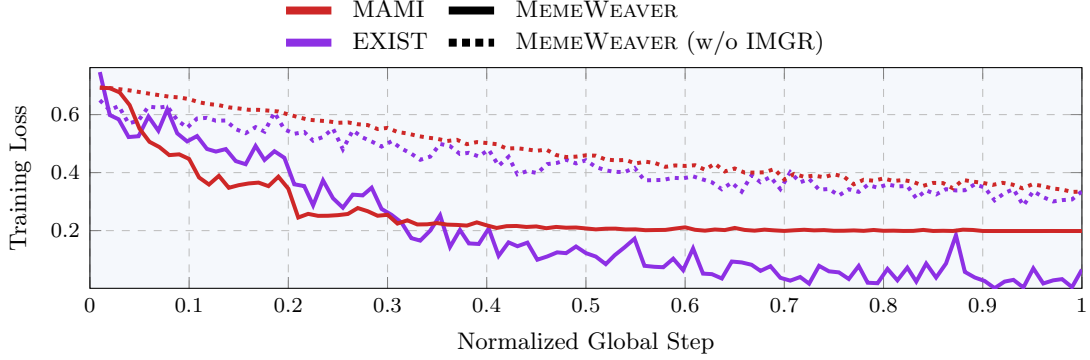


FIGURE 4.21. **Training loss evolution of MemeWeaver with and without IMGR across datasets.** Loss is plotted over normalized global training steps.

4.4.5 Results & Discussion

Performance Comparison Table 4.14 shows that MEMEWEAVER with IMGR consistently outperforms ablated variants without graph reasoning, particularly on MAMI and EXIST-English, with gains of 7-14 F1 points depending on the configuration, yielding more discriminative feature representations across memes. This effectiveness is further supported by Table 4.15, where we establish a new state of the art on MAMI over prior work.

Performance differences across datasets reflect the subtler, more complex nature of sexist discourse in EXIST compared to the more explicit, hate-driven memes in MAMI. This may also explain why incorporating complementary image captioning is more beneficial in the sexism case. Yet, the limited impact of semantic captioning (Prompt B) aligns with the weaker performance of our zero-shot LLM experiments, suggesting that even these powerful systems struggle with context-dependent social phenomena. This contrasts with prior studies that emphasized captioning as crucial for multimodal hate detection [Cao et al. \[2022, 2023\]](#). Furthermore, beyond improved overall performance, our method exhibits greater robustness across languages compared to LLMs. Specifically, LLMs show a substantial cross-lingual discrepancy, with an average F1 gap of 8.9 ± 2.7 between the English and Spanish EXIST partitions.

Overall, the results highlight the central role of learnable graph reasoning and robust multimodal fusion, which proved to adapt more effectively than both LLMs and hand-crafted heuristics.

Training Convergence Beyond boosting performance, the IMGR module in MEMEWEAVER substantially improves CLIP’s learning dynamics. As shown in Figure 4.21, the standard CLIP (dash lines) exhibits a roughly linear, gradual decrease in training loss over the full schedule. In contrast, MEMEWEAVER plunges almost immediately—especially on MAMI—halving its loss in under 20% of the training steps and reaching a stable plateau by 40%. Although the effect is less pronounced on the EXIST corpus, it still yields a

Fusion	MAMI			EXIST		
	Acc	F1	AUC	Acc	F1	AUC
Concat	73.6	73.5	82.0	76.3	71.6	75.7
MFB	77.6	77.4	83.4	<u>74.6</u>	<u>70.5</u>	<u>75.7</u>
GMU	<u>74.4</u>	<u>74.3</u>	<u>82.1</u>	71.6	60.3	73.6

TABLE 4.16. **Comparison of modality fusion strategies.** Best results are in bold and second-best are underlined.

faster descent and a lower steady-state loss than the vanilla model.

Modality Fusion Strategy Table 4.16 shows the impact of different fusion strategies on our best configurations across both datasets. On EXIST, naive concatenation performs surprisingly well, outperforming both MFB and GMU. In particular, GMU performs poorly, suggesting that gated mechanisms may be unstable in low-resource settings, where dynamic reweighting can amplify noise. In contrast, both MFB and GMU yield substantial gains on MAMI, with MFB achieving the best overall results. These findings highlight the data-dependent nature of fusion effectiveness and stress the need for systematic evaluation, rather than assuming that higher model complexity ensures better integration.

Inference Batch Size Batch size is a key hyperparameter, as it determines graph depth and the number of inter-instance relations. Figure 4.22 shows that F1 improves with larger inference-time batches, plateauing around 20. On MAMI, performance remains stable beyond this point, while on EXIST it fluctuates more, likely due to the smaller training set and greater sensitivity to this hyperparameter. However, the model stays competitive even with small batches: with size one, performance drops only slightly, especially on MAMI, despite the absence of inter-instance connections. We attribute this robustness to training, where MEMEWEAVER reaches a much lower loss minimum than when trained without IMGR (see Figure 4.21). Thus, even when reasoning cannot be fully exploited at inference time, the model continues to benefit from its stronger training dynamics.

Embedding Space Analysis To better understand the contribution of IMGR, we compare embeddings $\mathbf{f}$ (without) and $\mathbf{f}'$ (with IMGR). We first apply PCA [Abdi and Williams, 2010] to reduce embeddings to 50 dimensions, followed by t-SNE [Maaten and Hinton, 2008] for 2D projection. As shown in Figure 4.20, the $\mathbf{f}'$ embeddings exhibit clearer separation w.r.t. ground-truth labels. To quantify this effect, we train a linear logistic classifier on the 2D PCA→t-SNE embeddings. The 5-fold cross-validation accuracies, reported below each plot in Figure 4.20, confirm the superior discriminative power of $\mathbf{f}'$, supporting our hypothesis and explaining the performance gains of our variant.

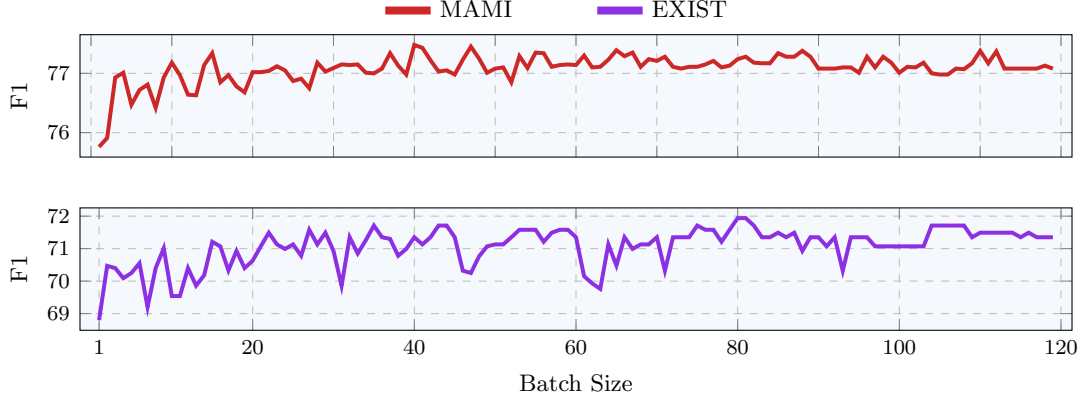


FIGURE 4.22. **F1 scores evolution as inference batch size** increases for MAMI (top) and EXIST (bottom).

Embedding Affinity Analysis To better understand the connections learned by IMGR, we examined the relationship between meme similarity and affinity scores R . Using fused multimodal embeddings $\mathbf{f}$, we computed pairwise cosine similarities across memes and grouped them by class alignment (both positive, both negative, or mixed). We deliberately avoided the refined embeddings $\mathbf{f}'$, as each incorporates information from all others, which would distort pairwise similarity estimates.

As shown in Figure 4.23, MAMI exhibits a strong positive Pearson correlation ($r = 0.819$): similar memes tend to receive stronger affinity scores, reinforcing their connections. This contrasts with the heuristic of Xu et al. [2025], which links dissimilar instances to maximize contrastive learning. EXIST, instead, aligns more closely with that heuristic, though with a weaker negative correlation ($r = -0.5937$). A post hoc test further confirmed the importance of these learned patterns: simply inverting affinity signs at inference time caused a severe performance drop (≈ 30 points), suggesting that graph structures must adapt to dataset characteristics rather than fixed heuristics.

We also find that positive pairs cluster in the upper half of the figure, indicating thematic coherence within classes and reflecting the social interplay. Finally, both datasets contain many meme pairs with affinity scores near zero, producing a sparse graph that avoids over-connecting unrelated memes and preserves meaningful distinctions.

4.4.6 Conclusion

We introduced MEMEWEAVER, a fully learnable graph-based framework for multimodal detection of sexism and misogyny in memes. Our method consistently outperforms the state of the art on challenging benchmarks, highlighting the benefits of batch-level graph-based reasoning and advanced multimodal fusion. Analyses of embedding spaces and affinity patterns further revealed meaningful structures and dataset-dependent behaviors, stressing the need to adapt modeling assumptions to context. These results open avenues for more nuanced relation modeling and fine-grained archetype catego-

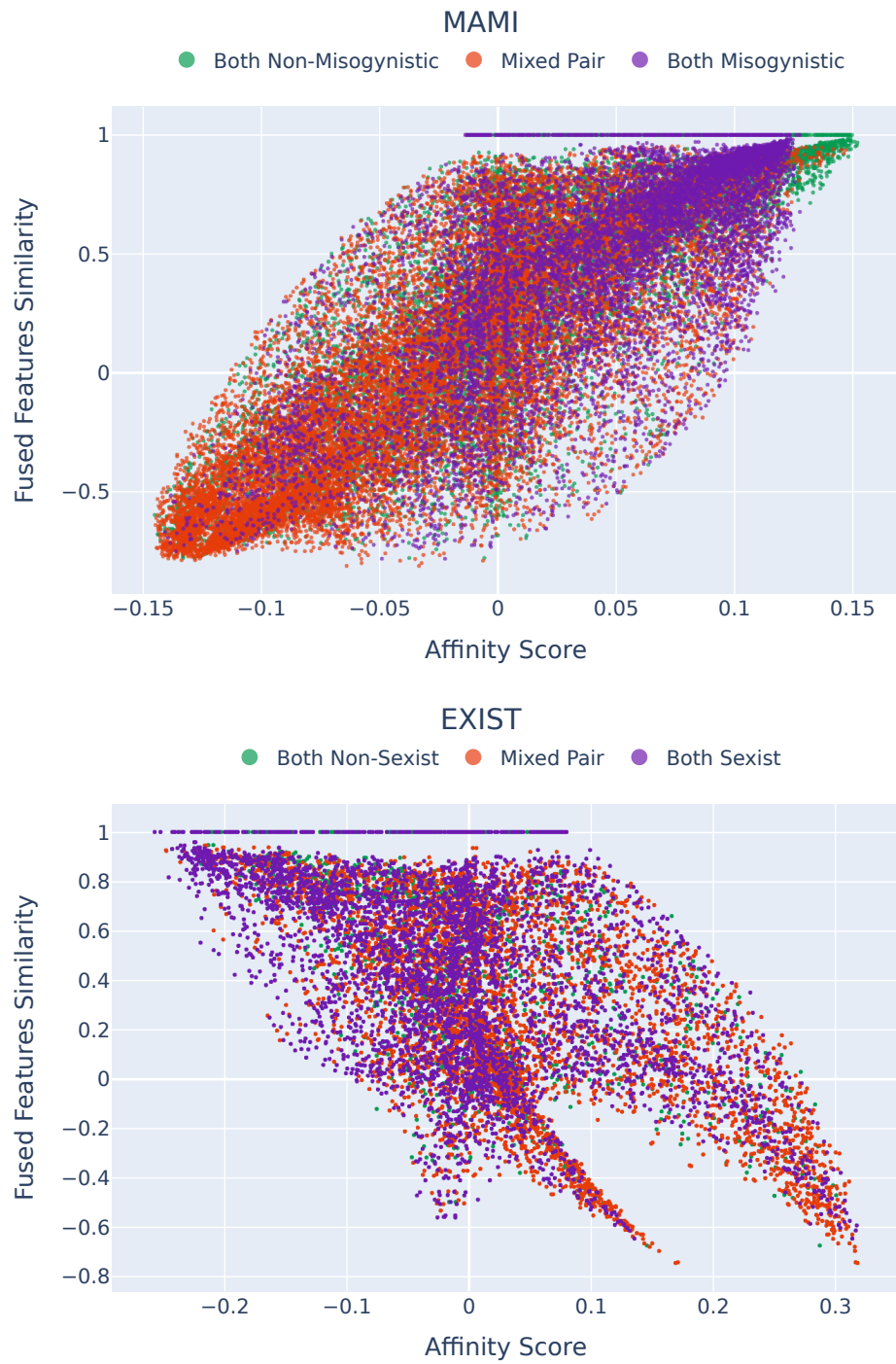


FIGURE 4.23. **Affinity patterns in IMGR.** Correlation between learned affinity scores and fused embedding similarity, with dataset-specific behaviors.

rization in multimodal hate speech detection [Rizzi et al., 2023], as well as extending MEMEWEAVER to broader hate detection tasks beyond sexism and misogyny.

Limitations

Despite its contributions, our work has several limitations that warrant further exploration. First, while we experimented with LLMs for image captioning, we did not integrate them as backbones for MEMEWEAVER, which could enhance semantic and contextual understanding; yet this was infeasible due to computational limits, particularly constraints on batch size, which is a critical hyperparameter of our proposed graph-based framework. Second, we evaluate large multimodal LLMs in a zero-shot setting, which is also their most common usage scenario; nevertheless, fine-tuned comparisons remain an interesting avenue for future research. Third, we only tested two prompt types for captioning (surface-level vs. semantic inference) and did not explore other prompt formats or reasoning chains, which may improve visual-textual grounding. Fourth, for EXIST-Spanish, we relied on machine translation, which we manually verified and found reliable; however, translation may still lose subtle cultural or linguistic cues, and future research should employ native multilingual encoders. Finally, the inter-meme graph is built within fixed training batches, which ensures efficiency but may restrict the capture of global structures or cross-batch relations; future work should investigate scalable or memory-augmented graph reasoning to improve global coherence.

Ethical Considerations

While MEMEWEAVER advances automated detection of harmful online content targeting women, the deployment of such systems raises concerns about potential biases that may disproportionately flag content from marginalized communities or fail to capture culturally-specific forms of sexism and misogyny. The use of automated moderation systems must be balanced with human oversight, as misclassification can lead to unjust content removal or allow harmful content to remain online.

4.4.7 Backbone Variant

Beyond CLIP, we explore a custom backbone for MEMEWEAVER using XLM-RoBERTa [Conneau et al., 2020] as the text encoder and Vision Transformer [Dosovitskiy et al., 2021] as the image encoder. Table 4.17 compares the XLM-R/ViT variant to our best-performing CLIP-based settings (no prompts for MAMI and Prompt A for EXIST; see Table 4.14). Although the custom backbone does not surpass the CLIP-based model, MEMEWEAVER continues to outperform standard baselines on most metrics (with the exception of F1 on EXIST), highlighting the framework’s robustness and adaptability across backbone architectures.

4.4.8 Zero-shot LLM prompts

Figure 4.24 reports the prompts used for zero-shot experiments using LLMs.

Backbone	Meme Weaver	MAMI			EXIST		
		Acc	F1	AUC	Acc	F1	AUC
CLIP	✗	71.9	71.8	79.4	68.2	59.5	75.1
	✓	77.6	77.4	83.4	76.3	73.6	72.9
XLM-R/ViT	✗	66.8	66.8	74.7	71.1	66.1	67.2
	✓	72.5	72.4	78.6	72.5	63.5	71.4

TABLE 4.17. **Performance of MemeWeaver with different backbone architectures.** Best results are in bold.

MAMI
Consider the following definitions. 1. ‘MISOGYNISTIC’ { ‘The meme expresses hatred, hostility, or contempt toward women, including insulting, demeaning, or promoting harm or discrimination against them.’ 2. ‘NON-MISOGYNISTIC’ - ‘The meme does not express hatred, hostility, or contempt, and does not promote harm, discrimination, or demeaning attitudes against women.’ Classify the input meme as ‘MISOGYNISTIC’ or ‘NON-MISOGYNISTIC’ based on the above definitions, considering the image as well as the extracted text from the image. Provide the answer as either ‘MISOGYNISTIC’ or ‘NON-MISOGYNISTIC’ only. Extracted text from the image: {ocr_text}.
EXIST
Consider the following definitions: 1. ‘SEXIST’ { \The meme is sexist itself, describes a sexist situation or criticizes a sexist behavior." 2. ‘NON-SEXIST’ { \The meme does not prejudice, underestimate, or discriminate women." Also consider the extracted text and the prompt-based caption: Extracted text: {ocr_text}. Prompt-based caption: {caption_text}. Classify the input meme as ‘SEXIST’ or ‘NON-SEXIST’ based on the above definitions considering the image and text. Provide only ‘SEXIST’ or ‘NON-SEXIST.’

FIGURE 4.24. Zero-shot LLM prompts across datasets.

Chapter 5

Conclusions

This thesis explored the multifaceted problem of optimizing large language models (LLMs) from both efficiency- and performance-oriented perspectives. At its core, this research aimed to reconcile two often competing goals: reducing the computational and environmental cost of language modeling, while preserving, or even enhancing, model capability, interpretability, and generalization. Across the works presented, we have shown that thoughtful interventions—spanning data, architecture, and optimization levels—can substantially improve the resource-consciousness and practical utility of LLMs.

Our first line of inquiry addressed computational efficiency through dynamic token pruning. We introduced PRUNEPERT, a token selection framework that augments pre-trained language models with a pruning layer capable of learning token importance in an end-to-end fashion. Unlike prior approaches, PrunePert eliminates the need for pre-processing or heuristic token ranking, achieving a near twofold inference speed-up while improving accuracy on multiple classification tasks. The qualitative analyses revealed task-dependent token selection patterns, offering a deeper understanding of linguistic redundancy and relevance in Transformer-based models. These findings reinforce the view that efficiency does not necessarily come at the cost of performance, but can instead emerge from more selective computation.

Building on the theme of selective reasoning, DEARWATSON extended the notion of distillation to the domain of abstractive dialogue summarization. By introducing an annotate-then-generate paradigm, DearWatson encouraged models to explicitly reason about relevance before producing summaries. This approach led to new state-of-the-art results on benchmark datasets, improving both semantic recall and factual faithfulness. Importantly, it also enhanced interpretability, as relevance markers provided transparent evidence for generated outputs. This work highlights that interpretability and efficiency can be synergistic when relevance-driven modeling guides generation.

Complementing these compression-oriented methods, we explored architectural refinement through semantic and structural augmentation. In our framework for biomedical summarization, we infused LLMs with domain-specific semantic graphs, integrating Abstract Meaning Representations (AMRs) and event structures into Transformer architectures. The results demonstrated substantial gains in informativeness, factuality, and

readability, validating that structured semantics can enhance generative quality without increasing model size. This supports the thesis that architectural sophistication—rather than scale alone—can improve reasoning and factual grounding. Future extensions toward logic-based representations and neuro-symbolic decoding could further align language generation with formal reasoning.

The thesis also investigated knowledge distillation as a path toward sustainable deployment of specialized models. ACE-ATTORNEY proposed a framework for distilling large models into compact specialized language models (SLMs) through the creation of synthetic datasets such as SYN-LEQA. These distilled models demonstrated competitive, and in some cases superior, performance to their larger teachers while dramatically reducing computational cost and carbon footprint. Moreover, by simulating retrieval-based question answering, we showed that selective generation improves both retrieval and generation quality, reinforcing the value of distillation as both a compression and alignment mechanism.

Finally, with MEMEWEAVER, we extended the scope of optimization beyond text, developing a graph-based multimodal reasoning framework for detecting sexism and misogyny in memes. This work underscored the adaptability of efficient architectures to complex, multimodal, socially relevant tasks. The observed structure in learned embeddings suggested that efficiency and interpretability principles—central to this thesis—also apply to cross-modal contexts, opening paths toward fine-grained reasoning and fairness in multimodal LLMs.

Taken together, the contributions of this thesis establish a unified perspective on LLM optimization: one that spans from efficiency-driven compression to performance-driven architectural enhancement, underpinned by an emphasis on interpretability, relevance, and structure. By showing that minor architectural and algorithmic interventions can yield disproportionate gains, this work challenges the prevailing assumption that progress in LLMs must rely solely on scale.

In the broader context, these findings advocate for a new generation of resource-aware and semantically principled language models, capable of maintaining competitiveness while aligning with practical constraints of computation, sustainability, and transparency. The interplay between dynamic sparsity, structured semantics, and distilled specialization outlined here offers a foundation for future research into models that are not only larger in capacity but smarter in design. As the field moves forward, the ideas developed in this thesis—selectivity, semantic awareness, and architectural economy—may serve as guiding principles for building the next wave of efficient, interpretable, and socially responsible LLMs.

Bibliography

- Hervé Abdi and Lynne J Williams. Principal component analysis. Wiley interdisciplinary reviews: computational statistics, 2(4):433–459, 2010.
- Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. arXiv preprint arXiv:2412.08905, 2024.
- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023.
- Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. Docbert: BERT for document classification. CoRR, abs/1904.08398, 2019.
- 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. doi: 10.18653/V1/2020.FINDINGS-EMNLP.66. URL <https://doi.org/10.18653/v1/2020.findings-emnlp.66>.
- Joshua Ainslie, James Lee-Thorp, Michiel de Jong, et al. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. arXiv preprint arXiv:2305.13245, 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>.
- Chenxin An, Ming Zhong, Yiran Chen, Danqing Wang, et al. Enhancing scientific papers summarization with citation graph. In AAAI, pages 12498–12506. AAAI Press, 2021a.
- Chenxin An, Ming Zhong, Yiran Chen, et al. Enhancing scientific papers summarization with citation graph. In AAAI 2021, Virtual Event, February 2-9, 2021, pages 12498–12506. AAAI Press, 2021b.
- 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 Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 7383–7390, 2020.
- Sotiris Anagnostidis, Dario Pavllo, Luca Biggio, et al. Dynamic context pruning for efficient and interpretable autoregressive transformers. In NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023, 2023.

- Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. Leveraging linguistic structure for open domain information extraction. In ACL (1), pages 344–354. ACL, 2015.
- Maria Anzovino, Elisabetta Fersini, and Paolo Rosso. Automatic Identification and Classification of Misogynistic Language on Twitter. In Proc. of NLDB, pages 57–64. Springer, 2018.
- Iván Arcos and Paolo Rosso. Sexism identification on tiktok: A multimodal ai approach with text, audio, and video. In Proc. of CLEF, pages 61–73. Springer, 2024.
- Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? 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 2654–2662, 2014. URL <https://proceedings.neurips.cc/paper/2014/hash/ea8fcd92d59581717e06eb187f10666d-Abstract.html>.
- Thomas Bachlechner, Bodhisattwa Prasad Majumder, Huanru Henry Mao, Gary Cottrell, et al. Rezero is all you need: fast convergence at large depth. In UAI, volume 161 of Proceedings of Machine Learning Research, pages 1352–1361. AUAI Press, 2021.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibor Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-v1 technical report. arXiv preprint arXiv:2502.13923, 2025.
- Vidhisha Balachandran, Artidoro Pagnoni, Jay Yoon Lee, Dheeraj Rajagopal, et al. Structsum: Summarization via structured representations. In EACL, pages 2575–2585. ACL, 2021.
- Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, et al. Abstract meaning representation for sembanking. In LAW@ACL, pages 178–186. ACL, 2013.
- Siqi Bao, Huang He, Fan Wang, Hua Wu, et al. PLATO: Pre-trained dialogue generation model with discrete latent variable. In ACL, pages 85–96, Online, July 2020. ACL. doi: 10.18653/v1/2020.acl-main.9. URL <https://aclanthology.org/2020.acl-main.9>.
- Alvaro Bartolome, Gabriel Martin, and Daniel Vila. Notus. <https://github.com/argilla-io/notus>, 2023.
- Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. CoRR, abs/2004.05150, 2020a.
- Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150, 2020b.
- Emily M. Bender and Alexander Koller. Climbing towards NLU: on meaning, form, and understanding in the age of data. In ACL, pages 5185–5198. ACL, 2020.
- Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In FAccT, pages 610–623. ACM, 2021.
- Quentin Berthet, Mathieu Blondel, Olivier Teboul, et al. Learning with differentiable perturbed optimizers. In NeurIPS, pages 9508–9519, 2020.
- Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. One SPRING to rule them both: Symmetric AMR semantic parsing and generation without a complex pipeline. In AAAI, pages 12564–12573. AAAI Press, 2021.
- Steven Bird, Ewan Klein, and Edward Loper. Natural language processing with Python: analyzing text with the natural language toolkit. O’Reilly Media, Inc.

- 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, 1979.
- 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. URL <https://aclanthology.org/2022.nllp-1.17>.
- Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. Inpars: Data augmentation for information retrieval using large language models. arXiv preprint arXiv:2202.05144, 2022.
- Jill C Bradley-Geist, Ivy Rivera, and Susan D Geringer. The Collateral Damage of Ambient Sexism: Observing Sexism Impacts Bystander Self-Esteem and Career Aspirations. Sex Roles, 73:29–42, 2015.
- Tom Brown, Benjamin Mann, Nick Ryder, et al. Language models are few-shot learners. NeurIPS, 33: 1877–1901, 2020.
- Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, et al. Sparks of artificial general intelligence: Early experiments with GPT-4. CoRR, abs/2303.12712, 2023. doi: 10.48550/arXiv.2303.12712. URL <https://doi.org/10.48550/arXiv.2303.12712>.
- Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In Tina Eliassi-Rad, Lyle H. Ungar, Mark Craven, and Dimitrios Gunopulos, editors, Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006, pages 535–541. ACM, 2006. doi: 10.1145/1150402.1150464. URL <https://doi.org/10.1145/1150402.1150464>.
- Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel Weld. TLDR: Extreme summarization of scientific documents. In EMNLP 2020, pages 4766–4777, Online, November 2020. ACL. doi: 10.18653/v1/2020.findings-emnlp.428.
- Shu Cai and Kevin Knight. Smatch: an evaluation metric for semantic feature structures. In ACL (2), pages 748–752. ACL, 2013.
- Rui Cao, Roy Ka-Wei Lee, Wen-Haw Chong, and Jing Jiang. Prompting for Multimodal Hateful Meme Classification. In Proc. of EMNLP, pages 321–332, 2022.
- Rui Cao, Ming Shan Hee, Adriel Kuek, Wen-Haw Chong, Roy Ka-Wei Lee, and Jing Jiang. Pro-Cap: Leveraging a Frozen Vision-Language Model for Hateful Meme Detection. In Proc. of the 31st ACM ICM, pages 5244–5252, 2023.
- Yu Cao, Wei Bi, Meng Fang, and Dacheng Tao. Pretrained language models for dialogue generation with multiple input sources. In EMNLP, pages 909–917, Online, November 2020. ACL. doi: 10.18653/v1/2020.findings-emnlp.81. URL <https://aclanthology.org/2020.findings-emnlp.81>.
- Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. Faithful to the original: Fact aware neural abstractive summarization. In AAAI, pages 4784–4791. AAAI Press, 2018.
- Aldo Gael Carranza, Reza Farahani, Natalia Ponomareva, Alex Kurakin, Matthew Jagielski, and Milad Nasr. Privacy-preserving recommender systems with synthetic query generation using differentially private large language models. arXiv preprint arXiv:2305.05973, 2023.
- Walter Cerroni, Gianluca Moro, Tommaso Pirini, and Marco Ramilli. Peer-to-peer data mining classifiers for decentralized detection of network attacks. In Hua Wang and Rui Zhang, editors, ADC, volume 137 of CRPIT, pages 101–108. ACS, 2013.

- Youngjin Chae and Thomas Davidson. Large language models for text classification: From zero-shot learning to fine-tuning. Open Science Foundation, 2023.
- Muthu Kumar et al. Chandrasekaran. Overview and insights from the shared tasks at scholarly document processing 2020: CL-SciSumm, LaySumm and LongSumm. In Proceedings of the First Workshop on Scholarly Document Processing, pages 214–224, Online, November 2020. ACL. doi: 10.18653/v1/2020.sdp-1.24.
- Jiaao Chen and Diyi Yang. Multi-view sequence-to-sequence models with conversational structure for abstractive dialogue summarization. In EMNLP, pages 4106–4118, Online, November 2020. ACL. doi: 10.18653/v1/2020.emnlp-main.336. URL <https://aclanthology.org/2020.emnlp-main.336>.
- Jiaao Chen and Diyi Yang. Structure-aware abstractive conversation summarization via discourse and action graphs. In NAACL, pages 1380–1391, Online, June 2021a. ACL. doi: 10.18653/v1/2021.naacl-main.109. URL <https://aclanthology.org/2021.naacl-main.109>.
- Jiaao Chen and Diyi Yang. Structure-aware abstractive conversation summarization via discourse and action graphs. In NAACL-HLT, pages 1380–1391. ACL, 2021b.
- Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. DialogSum: A real-life scenario dialogue summarization dataset. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 5062–5074, Online, August 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.449. URL <https://aclanthology.org/2021.findings-acl.449>.
- Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. Dialogsum: A real-life scenario dialogue summarization dataset. arXiv preprint arXiv:2105.06762, 2021b.
- Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019, pages 4793–4801. IEEE, 2019. doi: 10.1109/ICCV.2019.00489. URL <https://doi.org/10.1109/ICCV.2019.00489>.
- 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, 2022a. doi: 10.48550/ARXIV.2210.11416. URL <https://doi.org/10.48550/arXiv.2210.11416>.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, et al. Scaling instruction-finetuned language models. CoRR, abs/2210.11416, 2022b. doi: 10.48550/arXiv.2210.11416. URL <https://doi.org/10.48550/arXiv.2210.11416>.
- Christopher Clark, Kenton Lee, Ming-Wei Chang, et al. Boolq: Exploring the surprising difficulty of natural yes/no questions. In NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, pages 2924–2936. ACL, 2019.
- Alessio Cocchieri, Luca Ragazzi, Paolo Italiani, Giuseppe Tagliavini, and Gianluca Moro. “what do you call a dog that is incontrovertibly true? dogma”: Testing LLM generalization through humor. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 22922–22937, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1117. URL <https://aclanthology.org/2025.acl-long.1117/>.

- Arman Cohan, Franck Dernoncourt, Doo Soon Kim, et al. A discourse-aware attention model for abstractive summarization of long documents. In NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, pages 615–621. ACL, 2018.
- Meri Coleman and Ta Lin Liao. A computer readability formula designed for machine scoring. Journal of Applied Psychology, 60:283–284, 1975a.
- Meri Coleman and Ta Lin Liao. A computer readability formula designed for machine scoring. Journal of Applied Psychology, 60:283–284, 1975b.
- Charles Condevaux and Sébastien Harispe. LSG attention: Extrapolation of pretrained transformers to long sequences. In PAKDD, volume 13935, pages 443–454. Springer, 2023.
- Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann LeCun. Very deep convolutional networks for text classification. In EACL 2017, Valencia, Spain, April 3-7, 2017, pages 1107–1116. ACL, 2017.
- 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. Tetraault, 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>.
- Jean-Baptiste Cordonnier, Aravindh Mahendran, Alexey Dosovitskiy, et al. Differentiable patch selection for image recognition. In CVPR 2021, virtual, June 19-25, 2021, pages 2351–2360. Computer Vision Foundation / IEEE, 2021.
- 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>.
- Leyang Cui, Yu Wu, Shujie Liu, Yue Zhang, and Ming Zhou. Mutual: A dataset for multi-turn dialogue reasoning. arXiv preprint arXiv:2004.04494, 2020.
- Edgar Dale and Jeanne S. Chall. A formula for predicting readability. Educational Research Bulletin, 27(1):11–28, 1948.
- A de Paula, G Rizzi, E Fersini, and D Spina. AI-UPV at EXIST 2023–Sexism Characterization Using Large Language Models Under The Learning with Disagreements Regime. In CEUR WORKSHOP PROCEEDINGS, volume 3497, pages 985–999. CEUR-WS, 2023.
- 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, 2019a. doi: 10.18653/V1/N19-1423. URL <https://doi.org/10.18653/v1/n19-1423>.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. NAACL-HLT, pages 4171–4186, 2019b.

- Jay DeYoung, Iz Beltagy, Madeleine van Zuylen, Bailey Kuehl, et al. MS²: Multi-document summarization of medical studies. In EMNLP, pages 7494–7513. ACL, November 2021.
- Shibhansh Dohare, Vivek Gupta, and Harish Karnick. Unsupervised semantic abstractive summarization. In ACL (3), pages 74–83. ACL, 2018.
- Giacomo Domeniconi, Gianluca Moro, Roberto Pasolini, and Claudio Sartori. Cross-domain text classification through iterative refining of target categories representations. In International Conference on Knowledge Discovery and Information Retrieval, volume 2, pages 31–42. SciTePress, 2014.
- Giacomo Domeniconi, Gianluca Moro, Andrea Pagliarani, and Roberto Pasolini. Markov chain based method for in-domain and cross-domain sentiment classification. pages 127–137. SciTePress, 2015a. doi: 10.5220/0005636001270137.
- Giacomo Domeniconi, Gianluca Moro, Roberto Pasolini, and Claudio Sartori. Iterative refining of category profiles for nearest centroid cross-domain text classification. In Knowledge Discovery, Knowledge Engineering and Knowledge Management: 6th International Joint Conference, IC3K 2014, Rome, Italy, October 21-24, 2014, Revised Selected Papers 6, pages 50–67. Springer, 2015b.
- Giacomo Domeniconi, Konstantinos Semertzidis, Vanessa Lopez, Elizabeth M Daly, Spyros Kotoulas, and Gianluca Moro. A novel method for unsupervised and supervised conversational message thread detection. In International Conference on Data Management Technologies and Applications, volume 2, pages 43–54. SciTePress, 2016.
- Giacomo Domeniconi, Gianluca Moro, Andrea Pagliarani, Roberto Pasolini, et al. On deep learning in cross-domain sentiment classification. In Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management-(Volume 1), volume 1, pages 50–60. SciTePress, 2017.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. 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=YicbFdNTTy>.
- Ari Drucker, Ofer Fein, Dafna Bergerbest, and Rachel Giora. On Sarcasm, Social Awareness, and Gender. Humor, 27(4):551–573, 2014.
- Maeve Duggan. Online Harassment. https://www.pewresearch.org/internet/wp-content/uploads/sites/9/2017/07/PI_2017.07.11_Online-Harassment_FINAL.pdf, 2017.
- Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, et al. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In ACL, pages 1074–1084, Florence, Italy, July 2019. ACL. doi: 10.18653/v1/P19-1102. URL <https://aclanthology.org/P19-1102>.
- Guglielmo Faggioli, Laura Dietz, Charles LA Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Benno Stein, et al. Perspectives on large language models for relevance judgment. In Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval, pages 39–50, 2023.
- Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. Using local knowledge graph construction to scale seq2seq models to multi-document inputs. In EMNLP/IJCNLP (1), pages 4184–4194. ACL, 2019.
- Xiachong Feng, Xiaocheng Feng, Libo Qin, Bing Qin, et al. Language model as an annotator: Exploring DialoGPT for dialogue summarization. In ACL-IJCNLP, pages 1479–1491, Online, August 2021. ACL. doi: 10.18653/v1/2021.acl-long.117. URL <https://aclanthology.org/2021.acl-long.117>.

- Xiachong Feng, Xiaocheng Feng, and Bing Qin. A survey on dialogue summarization: Recent advances and new frontiers. In *IJCAI*, pages 5453–5460. ijcai.org, 2022.
- Elisabetta Fersini, Francesca Gasparini, Giulia Rizzi, Aurora Saibene, Berta Chulvi, Paolo Rosso, Alyssa Lees, and Jeffrey Sorensen. SemEval-2022 Task 5: Multimedia Automatic Misogyny Identification. In *Proc. of the 16th SemEval@NAACL*, pages 533–549. ACL, 2022. doi: <https://doi.org/10.18653/v1/2022.semeval-1.74>.
- Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *CoRR*, abs/1903.02428, 2019.
- Mauajama Firdaus, Arunav Pratap Shandeelya, and Asif Ekbal. More to diverse: Generating diversified responses in a task oriented multimodal dialog system. *PLoS ONE*, 15, 2020. URL <https://api.semanticscholar.org/CorpusID:226269766>.
- Lara Fontanella, Berta Chulvi, Elisa Ignazzi, Annalina Sarra, and Alice Tontodimamma. How do we study misogyny in the digital age? A systematic literature review using a computational linguistic approach. *Humanities and Social Sciences Communications*, 11(1):1–15, 2024.
- Giacomo Frisoni and Gianluca Moro. Phenomena explanation from text: Unsupervised learning of interpretable and statistically significant knowledge. In *Data Management Technologies and Applications: 9th International Conference, DATA 2020, Virtual Event, July 7–9, 2020, Revised Selected Papers 9*, pages 293–318. Springer, 2021.
- Giacomo Frisoni, Gianluca Moro, Antonella Carbonaro, et al. Learning interpretable and statistically significant knowledge from unlabeled corpora of social text messages: A novel methodology of descriptive text mining. In *DATA*, pages 121–132, 2020.
- 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.
- Giacomo Frisoni, Antonella Carbonaro, Gianluca Moro, Andrea Zammarchi, and Marco Avagnano. NLG-metricverse: An end-to-end library for evaluating natural language generation. In *COLING*, pages 3465–3479, Gyeongju, Republic of Korea, October 2022. COLING. 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 *DATA*, pages 168–179. INSTICC, SciTePress, 2022.
- Giacomo Frisoni, Miki Mizutani, Gianluca Moro, and Lorenzo Valgimigli. Bioreader: a retrieval-enhanced text-to-text transformer for biomedical literature. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 5770–5793, 2022a.
- Giacomo Frisoni, Gianluca Moro, and Lorenzo Balzani. Text-to-text extraction and verbalization of biomedical event graphs. In *COLING*, pages 2692–2710, Gyeongju, Republic of Korea, October 2022b. COLING. 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, 2022c.
- 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 *AAAI*, pages 1–9. AAAI Press, 2023a.

- 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 Proceedings of the AAAI conference on artificial intelligence, volume 37, pages 12781–12789, 2023b.
- Tommaso Furlanello, Zachary Chase Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born-again neural networks. In Jennifer G. Dy and Andreas Krause, editors, Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, volume 80 of Proceedings of Machine Learning Research, pages 1602–1611. PMLR, 2018. URL <http://proceedings.mlr.press/v80/furlanello18a.html>.
- 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. URL https://openreview.net/pdf?id=h50pjGd_lo6.
- Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, et al. Dialogue response ranking training with large-scale human feedback data. In EMNLP, pages 386–395, Online, November 2020. ACL. doi: 10.18653/v1/2020.emnlp-main.28. URL <https://aclanthology.org/2020.emnlp-main.28>.
- Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. In The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024. OpenReview.net, 2024.
- Sreyan Ghosh, Chandra Kiran Reddy Evuru, Sonal Kumar, Ramaneswaran S, S. Sakshi, Utkarsh Tyagi, and Dinesh Manocha. DALE: generative data augmentation for low-resource legal NLP. 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 8511–8565. Association for Computational Linguistics. URL <https://aclanthology.org/2023.emnlp-main.528>.
- 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.
- Paul Giglioli, Nikhita Sagar, Anand S. Rao, and Joseph Voyles. Domain-aware abstractive text summarization for medical documents. In BIBM, pages 2338–2343. IEEE Comp. Society, 2018.
- Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. Chatgpt outperforms crowd-workers for text-annotation tasks. CoRR, abs/2303.15056, 2023a. doi: 10.48550/arXiv.2303.15056. URL <https://doi.org/10.48550/arXiv.2303.15056>.
- Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. Chatgpt outperforms crowd workers for text-annotation tasks. Proceedings of the National Academy of Sciences, 120(30), July 2023b.
- Peter Glick and Susan T Fiske. Hostile and Benevolent Sexism as Complementary Justifications for Gender Inequality. American psychologist, 56(2):109, 2001.
- Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In Proceedings of the 2nd Workshop on New Frontiers in Summarization, pages 70–79, Hong Kong, China, November 2019. ACL. doi: 10.18653/v1/D19-5409. URL <https://aclanthology.org/D19-5409>.
- Raul Gomez, Jaume Gibert, Lluís Gomez, and Dimosthenis Karatzas. Exploring Hate Speech Detection in Multimodal Publications. In Proc. of the IEEE/CVF WACV, pages 1470–1478, 2020. doi: 10.1109/WACV45572.2020.9093414.
- Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, et al. Power-bert: Accelerating bert inference via progressive word-vector elimination. In ICML, pages 3690–3699, 2020.

- Jia-Chen Gu, Chongyang Tao, Zhenhua Ling, Can Xu, et al. MPC-BERT: A pre-trained language model for multi-party conversation understanding. In *ACL*, pages 3682–3692, Online, August 2021. ACL. doi: 10.18653/v1/2021.acl-long.285. URL <https://aclanthology.org/2021.acl-long.285>.
- Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lifeng Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Gang Wang, Liang Wang, Xinchao Wang, et al. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, 2018. doi: 10.1016/j.patcog.2017.10.013.
- R.; et al Gunning. *Technique of clear writing*. 1952.
- Yue Guo, Wei Qiu, Yizhong Wang, and Trevor Cohen. Automated lay language summarization of biomedical scientific reviews. In *AAAI 2021, Virtual Event, February 2-9, 2021*, pages 160–168. AAAI Press, 2021a.
- Yue Guo, Wei Qiu, Yizhong Wang, and Trevor Cohen. Automated lay language summarization of biomedical scientific reviews. In *AAAI*, pages 160–168. AAAI Press, 2021b.
- Zhen Guo, Peiqi Wang, Yanwei Wang, and Shangdi Yu. Dr. llama: Improving small language models in domain-specific qa via generative data augmentation. *arXiv preprint arXiv:2305.07804*, 2023.
- Hardy and Andreas Vlachos. Guided neural language generation for abstractive summarization using abstract meaning representation. In *EMNLP*, pages 768–773. ACL, 2018.
- Serhii Havrylov and Ivan Titov. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. 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 2149–2159, 2017. URL <https://proceedings.neurips.cc/paper/2017/hash/70222949cc0db89ab32c9969754d4758-Abstract.html>.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: decoding-enhanced bert with disentangled attention. In *ICLR*. OpenReview.net, 2021. URL <https://openreview.net/forum?id=XPZIaotutsD>.
- Liam Hebert, Gaurav Sahu, Yuxuan Guo, Nanda Kishore Sreenivas, Lukasz Golab, and Robin Cohen. Multi-Modal Discussion Transformer: Integrating Text, Images and Graph Transformers to Detect Hate Speech on Social Media. In *Proc. of AAAI*, volume 38, pages 22096–22104, 2024.
- Michael Herz and Péter Molnár. *The Content and Context of Hate Speech: Rethinking Regulation and Responses*. Cambridge University Press, 2012.
- 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>.
- Gordon Hodson, Jonathan Rush, and Cara C MacInnis. A Joke is Just a Joke (Except When It Isn’t): Cavalier Humor Beliefs Facilitate the Expression of Group Dominance Motives. *Journal of personality and social psychology*, 99(4):660, 2010.
- 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. URL <https://ceur-ws.org/Vol-2645/paper5.pdf>.

- Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength natural language processing in python. 2020.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, et al. Lora: Low-rank adaptation of large language models. In ICLR. OpenReview.net, 2022. URL <https://openreview.net/forum?id=nZeVKeeFYf9>.
- Fan Huang, Haewoon Kwak, and Jisun An. Is chatgpt better than human annotators? potential and limitations of chatgpt in explaining implicit hate speech. In ACM web conference, pages 294–297, 2023a.
- Kung-hsiang Huang, Siffi Singh, Xiaofei Ma, Wei Xiao, et al. SWING: Balancing coverage and faithfulness for dialogue summarization. In EACL, pages 512–525, Dubrovnik, Croatia, May 2023b. ACL. URL <https://aclanthology.org/2023.findings-eacl.37>.
- Luyang Huang, Lingfei Wu, and Lu Wang. Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward. In ACL, pages 5094–5107. ACL, 2020.
- Luyang Huang, Shuyang Cao, Nikolaus Nova Parulian, et al. Efficient attentions for long document summarization. In NAACL-HLT 2021, Online, June 6-11, 2021, pages 1419–1436. ACL, 2021.
- Xin Huang, Ashish Khetan, Rene Bidart, and Zohar Karnin. Pyramid-bert: Reducing complexity via successive core-set based token selection. arXiv preprint arXiv:2203.14380, 2022a.
- Xin Huang, Ashish Khetan, Rene Bidart, et al. Pyramid-bert: Reducing complexity via successive core-set based token selection. In ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 8798–8817. ACL, 2022b.
- 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>.
- Iris A. M. Huijben, Wouter Kool, Max B. Paulus, and Ruud J. G. van Sloun. A review of the gumbel-max trick and its extensions for discrete stochasticity in machine learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45:1353–1371, 2021. URL <https://api.semanticscholar.org/CorpusID:238259238>.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. arXiv preprint arXiv:2410.21276, 2024.
- Oana Ignat, Jing Xu, Xiang Ren, et al. Scaling up language models: Hardware, efficiency, and emerging capabilities. Transactions on Machine Learning Research, 1:1–18, 2023.
- Paolo Italiani, Giacomo Frisoni, Gianluca Moro, Antonella Carbonaro, and Claudio Sartori. Evidence, my dear watson: Abstractive dialogue summarization on learnable relevant utterances. Neurocomputing, 572:127132, 2024.
- Paolo Italiani, Gianluca Moro, and Luca Ragazzi. Enhancing legal question answering with data generation and knowledge distillation from large language models: P. italiani et al. Artificial Intelligence and Law, pages 1–26, 2025.
- Paolo Italiani, David Gimeno-Gomez, Luca Ragazzi, Gianluca Moro, and Paolo Rosso. Memeweaver: Inter-meme graph reasoning for sexism and misogyny detection, 2026a. URL <https://arxiv.org/abs/2601.08684>.

- Paolo Italiani, Luca Ragazzi, and Gianluca Moro. Read between the tokens: Differentiable text pruning via perturbed top-k selection. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2026b.
- Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In ICLR. OpenReview.net, 2017a. URL <https://openreview.net/forum?id=rkE3y85ee>.
- Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In ICLR (Poster). OpenReview.net, 2017b.
- Akshita Jha and Radhika Mamidi. When Does a Compliment Become Sexist? Analysis and Classification of Ambivalent Sexism using Twitter Data. In Proc. of the Workshop on NLP and Computational Social Science, pages 7–16, 2017.
- Jiaming Ji, Mickel Liu, Josef Dai, et al. Beavertails: Towards improved safety alignment of LLM via a human-preference dataset. In NeurIPS, 2023.
- Xin Ji and Wen Zhao. SKGSUM: abstractive document summarization with semantic knowledge graphs. In IJCNN, pages 1–8. IEEE, 2021.
- Qi Jia, Yizhu Liu, Haifeng Tang, and Kenny Zhu. Post-training dialogue summarization using pseudo-paraphrasing. In Findings of the Association for Computational Linguistics: NAACL 2022, pages 1660–1669, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-naacl.125. URL <https://aclanthology.org/2022.findings-naacl.125>.
- Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, et al. Mistral 7b. arXiv preprint arXiv:2310.06825, 2023a.
- Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmllingua: Compressing prompts for accelerated inference of large language models. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023, pages 13358–13376. Association for Computational Linguistics, 2023b.
- Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmllingua: Compressing prompts for accelerated inference of large language models. arXiv preprint arXiv:2310.05736, 2023c.
- Yong Jianhong. Token pruning for efficient nlp, vision, and speech models. 2025.
- 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. doi: 10.18653/V1/2020.FINDINGS-EMNLP.372. URL <https://doi.org/10.18653/v1/2020.findings-emnlp.372>.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. IEEE Transactions on Big Data, 7(3):535–547, 2019.
- Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomáš Mikolov. Bag of tricks for efficient text classification. In EACL, Valencia, Spain, pages 427–431. ACL, 2017.
- Pavan Kapanipathi, Ibrahim Abdelaziz, Srinivas Ravishankar, Salim Roukos, et al. Leveraging abstract meaning representation for knowledge base question answering. In ACL/IJCNLP (Findings), volume ACL/IJCNLP 2021 of Findings of ACL, pages 3884–3894. ACL, 2021.

- Nikiforos Karamanis. Book reviews: Text mining for biology and biomedicine, edited by sophia ananiadou and john McNaught. Computational Linguistics, 33(1):135–140, 2007. URL <https://aclanthology.org/J07-1007>.
- Maurice G Kendall. A new measure of rank correlation. Biometrika, 30(1/2):81–93, 1938.
- Zaid Khan and Yun Fu. Exploiting BERT for Multimodal Target Sentiment Classification through Input Space Translation. In Proc. of ACM ICM, pages 3034–3042, 2021.
- 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. doi: 10.18653/V1/2022.FINDINGS-ACL.50. URL <https://doi.org/10.18653/v1/2022.findings-acl.50>.
- Gyuwan Kim and Kyunghyun Cho. Length-adaptive transformer: Train once with length drop, use anytime with search. arXiv preprint arXiv:2010.07003, 2020.
- Jin-Hwa Kim, Kyoung Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Hadamard product for low-rank bilinear pooling. In Proc. of ICLR. OpenReview.net, 2017. URL <https://openreview.net/forum?id=r1rhWnZkg>.
- Mi-Young Kim, Randy Goebel, Yoshinobu Kano, and Ken Satoh. Coliee-2016: evaluation of the competition on legal information extraction and entailment. In International Workshop on Juris-informatics (JURISIN 2016), 2016.
- Sehoon Kim, Sheng Shen, David Thorsley, Amir Gholami, Woosuk Kwon, Joseph Hassoun, and Kurt Keutzer. Learned token pruning for transformers. In Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining, pages 784–794, 2022.
- Yeachen Kim, Junho Kim, Jun-Hyung Park, Mingyu Lee, and SangKeun Lee. Leap-of-thought: Accelerating transformers via dynamic token routing. 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 15757–15769. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.976. URL <https://doi.org/10.18653/v1/2023.emnlp-main.976>.
- Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In Jian Su, Kevin Duh, and Xavier Carreras, editors, Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1317–1327, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1139. URL <https://aclanthology.org/D16-1139/>.
- J. Peter Kincaid, Robert P. Fishburne, R L Rogers, and Brad S. Chissom. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. 1975a.
- J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, et al. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. 1975b.
- Jia Jin Koay, Alexander Roustai, Xiaojin Dai, Dillon Burns, et al. How domain terminology affects meeting summarization performance. In COLING, pages 5689–5695, Barcelona, Spain (Online), December 2020. COLING. doi: 10.18653/v1/2020.coling-main.499. URL <https://aclanthology.org/2020.coling-main.499>.
- Philipp Koehn. Statistical significance tests for machine translation evaluation. In Proceedings of the 2004 conference on empirical methods in natural language processing, pages 388–395, 2004.

- Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Xuan Shen, Geng Yuan, Bin Ren, Hao Tang, et al. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In European conference on computer vision, pages 620–640. Springer, 2022.
- Wouter Kool, Herke van Hoof, and Max Welling. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. 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 3499–3508. PMLR, 2019. URL <http://proceedings.mlr.press/v97/kool19a.html>.
- Robin M Kowalski and Elizabeth Whittaker. Cyberbullying: Prevalence, Causes, and Consequences. The Wiley handbook of Psychology, Technology, and Society, pages 142–157, 2015.
- Wojciech Kryscinski, Romain Paulus, Caiming Xiong, and Richard Socher. Improving abstraction in text summarization. In EMNLP, pages 1808–1817. ACL, 2018.
- 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, EMNLP, pages 9332–9346. ACL, 2020. doi: 10.18653/v1/2020.emnlp-main.750. URL <https://doi.org/10.18653/v1/2020.emnlp-main.750>.
- Varun Kumar, Ashutosh Choudhary, and Eunah Cho. Data augmentation using pre-trained transformer models. arXiv preprint arXiv:2003.02245, 2020.
- Tuan M Lai, Giuseppe Castellucci, Saar Kuzi, Heng Ji, and Oleg Rokhlenko. External knowledge acquisition for end-to-end document-oriented dialog systems. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 3633–3647, 2023.
- Man Lan, Zhihua Zhang, Yue Lu, and Ju Wu. Three convolutional neural network-based models for learning sentiment word vectors towards sentiment analysis. In IJCNN 2016, Vancouver, BC, Canada, July 24-29, 2016, pages 3172–3179. IEEE, 2016.
- Fei-Tzin Lee, Chris Kedzie, Nakul Verma, and Kathleen R. McKeown. An analysis of document graph construction methods for AMR summarization. CoRR, abs/2111.13993, 2021a.
- Kenton Lee, Kelvin Guu, Luheng He, Tim Dozat, and Hyung Won Chung. Neural data augmentation via example extrapolation. arXiv preprint arXiv:2102.01335, 2021b.
- 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. doi: 10.18653/V1/2020.ACL-MAIN.703. URL <https://doi.org/10.18653/v1/2020.acl-main.703>.
- 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. Proceedings of ACL, pages 7871–7880, 2020a.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, et al. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In ACL, pages 7871–7880. ACL, 2020b.

- Mike Lewis, Yinhan Liu, Naman Goyal, et al. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In ACL, July 5-10, 2020, pages 7871–7880. ACL, 2020c.
- Patrick 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, 2021.
- Junyi Li, Tianyi Tang, Wayne Xin Zhao, et al. Pre-trained language models for text generation: A survey. ACM Computing Surveys, 56(9):1–39, 2024a.
- Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. Visual Semantic Reasoning for Image-Text Matching. In Proc. of the IEEE/CVF ICCV, pages 4654–4662, 2019a.
- Manling Li, Lingyu Zhang, Heng Ji, and Richard J. Radke. Keep meeting summaries on topic: Abstractive multi-modal meeting summarization. In ACL, pages 2190–2196, Florence, Italy, July 2019b. ACL. doi: 10.18653/v1/P19-1210. URL <https://aclanthology.org/P19-1210>.
- Qian Li, Lixin Su, Jiashu Zhao, Long Xia, Hengyi Cai, Suqi Cheng, Hengzhu Tang, Junfeng Wang, and Dawei Yin. Text-Video Retrieval via Multi-Modal Hypergraph Networks. In Proc. of the ACM ICWSDM, pages 369–377, 2024b.
- Wei Li, Xinyan Xiao, Jiachen Liu, Hua Wu, et al. Leveraging graph to improve abstractive multi-document summarization. In ACL, pages 6232–6243. ACL, 2020.
- Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. Dailymdialog: A manually labelled multi-turn dialogue dataset. arXiv preprint arXiv:1710.03957, 2017.
- 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.
- Hui Lin and Vincent Ng. Abstractive summarization: A survey of the state of the art. In AAAI, pages 9815–9822. AAAI Press, 2019.
- Wei Liu, Haozhao Wang, et al. FR: folded rationalization with a unified encoder. In NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022, 2022.
- Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. Proceedings of EMNLP-IJCNLP, pages 3730–3740, 2019a.
- Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. In EMNLP/IJCNLP (1), pages 3728–3738. ACL, 2019b.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, et al. Roberta: A robustly optimized BERT pretraining approach. CoRR, abs/1907.11692, 2019a. URL <http://arxiv.org/abs/1907.11692>.
- Zhengyuan Liu and Nancy Chen. Controllable neural dialogue summarization with personal named entity planning. In ACL, pages 92–106, Online and Punta Cana, Dominican Republic, November 2021. ACL. doi: 10.18653/v1/2021.emnlp-main.8. URL <https://aclanthology.org/2021.emnlp-main.8>.
- Zhengyuan Liu, Angela Ng, Sheldon Lee Shao Guang, Ai Ti Aw, et al. Topic-aware pointer-generator networks for summarizing spoken conversations. In ASRU, pages 814–821. IEEE, 2019b.
- Zhengyuan Liu, Ke Shi, and Nancy Chen. Coreference-aware dialogue summarization. In Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 509–519, Singapore and Online, July 2021. ACL. URL <https://aclanthology.org/2021.sigdial-1.53>.

- 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, (ADC 2010), volume 104 of CRPIT, pages 171–178. Australian Computer Society, 2010a.
- 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, ADC, volume 104 of CRPIT, pages 171–178. ACS, 2010b.
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In ICLR. OpenReview.net. URL <https://openreview.net/forum?id=Bkg6RiCqY7>.
- Antoine Louis, Gijs van Dijk, 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>.
- Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, et al. Neurologic decoding: (un)supervised neural text generation with predicate logic constraints. In NAACL-HLT, pages 4288–4299. ACL, 2021.
- Kaicheng Luo, Huaxiong Li, Xianzhong Zhou, and Bing Huang. An attention-based token pruning method for vision transformers. In IJCRS 2022, Suzhou, China, November 11-14, 2022, Proceedings, volume 13633, pages 274–288. Springer, 2022a.
- Kaicheng Luo, Huaxiong Li, Xianzhong Zhou, and Bing Huang. An attention-based token pruning method for vision transformers. In International Joint Conference on Rough Sets, pages 274–288. Springer, 2022b.
- Xuan Luo, Bin Liang, Qianlong Wang, Jing Li, Erik Cambria, Xiaojun Zhang, Yulan He, Min Yang, and Ruifeng Xu. A Literature Survey on Multimodal and Multilingual Sexism Detection. IEEE Trans. on Computational Social Systems, pages 1–19, 2025. doi: 10.1109/TCSS.2025.3561921.
- Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. Journal of machine learning research, 9(Nov):2579–2605, 2008.
- Sean MacAvaney and Luca Soldaini. One-shot labeling for automatic relevance estimation. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 2230–2235, 2023.
- Andres Mafra, Sounak Dey, Ali Furkan Biten, Lluís Gomez, and Dimosthenis Karatzas. Multi-Modal Reasoning Graph for Scene-Text based Fine-Grained Image Classification and Retrieval. In Proc. of the IEEE/CVF WACV, pages 4023–4033, 2021.
- Kaleel Mahmood, Rigel Mahmood, and Marten Van Dijk. On the robustness of vision transformers to adversarial examples. In Proceedings of the IEEE/CVF international conference on computer vision, pages 7838–7847, 2021.
- Amir M. Mansourian, Rozhan Ahmadi, Masoud Ghafouri, Amir Mohammad Babaei, Elaheh Badali Golezani, Zeynab Yasamani Ghamchi, Vida Ramezani, Alireza Taherian, Kimia Dinashi, Amirali Miri, and Shohreh Kasaei. A comprehensive survey on knowledge distillation. Trans. Mach. Learn. Res., 2025, 2025. URL <https://openreview.net/forum?id=3cbJzdR78B>.
- Jorge Martinez-Gil. A survey on legal question–answering systems. Computer Science Review, 48: 100552, may 2023. doi: 10.1016/j.cosrev.2023.100552. URL <https://doi.org/10.1016%2Fj.cosrev.2023.100552>.
- Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan T. McDonald. On faithfulness and factuality in abstractive summarization. In ACL, pages 1906–1919. ACL, 2020.

- Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. Generating training data with language models: Towards zero-shot language understanding. *Advances in Neural Information Processing Systems*, 35: 462–477, 2022.
- Rada Mihalcea and Paul Tarau. Texttrank: Bringing order into text. In *EMNLP*, pages 404–411. ACL, 07 2004.
- Sewon Min, Mike Lewis, and Luke Zettlemoyer. Recent advances in natural language processing with deep learning. *Journal of Artificial Intelligence Research*, 79:1–35, 2024. doi: 10.1613/jair.1.13849.
- Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andrés Códas, 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>.
- Ali Modarressi, Hosein Mohebbi, and Mohammad Taher Pilehvar. Adapler: Speeding up inference by adaptive length reduction. In *ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1–15. ACL, 2022a.
- Ali Modarressi, Hosein Mohebbi, and Mohammad Taher Pilehvar. Adapler: Speeding up inference by adaptive length reduction. *arXiv preprint arXiv:2203.08991*, 2022b.
- Milad Moradi and Nasser Ghadiri. Text summarization in the biomedical domain. *arXiv preprint arXiv:1908.02285*, 2019.
- 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.
- 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, 2012a.
- 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. *Journal of Network and Computer Applications*, 35(4):1218–1234, 2012b.
- Gianluca Moro and Luca Ragazzi. Semantic self-segmentation for abstractive summarization of long documents in low-resource regimes. In *AAAI 2022, February 22 - March 1, 2022*, pages 11085–11093. AAAI Press, 2022a.
- Gianluca Moro and Luca Ragazzi. Semantic self-segmentation for abstractive summarization of long documents in low-resource regimes. In *AAAI*, pages 11085–11093. AAAI Press, 2022b.
- Gianluca Moro and Luca Ragazzi. Align-then-abstract representation learning for low-resource summarization. *Neurocomputing*, 548:126356, 2023.
- Gianluca Moro, Andrea Pagliarani, Roberto Pasolini, and Claudio Sartori. Cross-domain & In-domain Sentiment Analysis with Memory-based Deep Neural Networks. In *IC3K*, volume 1, pages 127–138. SciTePress, 2018a. doi: 10.5220/0007239101270138.
- Gianluca Moro, Andrea Pagliarani, Roberto Pasolini, Claudio Sartori, et al. Cross-domain & in-domain sentiment analysis with memory-based deep neural networks. In *Proceedings of the 10th International Joint Conference on KnowledgeDiscovery, Knowledge Engineering and Knowledge Management, IC3K2018, Volume 1: KDIR, Seville, Spain, September 18-20, 2018.*, pages 125–136. SciTePress, 2018b.

- Gianluca Moro, Luca Ragazzi, Lorenzo Valgimigli, and Davide Freddi. Discriminative marginalized probabilistic neural method for multi-document summarization of medical literature. In ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 180–189. ACL, 2022a.
- Gianluca Moro, Luca Ragazzi, Lorenzo Valgimigli, and Davide Freddi. Discriminative marginalized probabilistic neural method for multi-document summarization of medical literature. In ACL (1), pages 180–189. ACL, 2022b.
- Gianluca Moro, Luca Ragazzi, and Lorenzo Valgimigli. Carburacy: Summarization models tuning and comparison in eco-sustainable regimes with a novel carbon-aware accuracy. In AAAI 2023, Washington, DC, USA, February 7-14, 2023, pages 14417–14425. AAAI Press, 2023a.
- Gianluca Moro, Luca Ragazzi, and Lorenzo Valgimigli. Graph-based abstractive summarization of extracted essential knowledge for low-resource scenarios. In ECAI 2023, September 30 - October 4, 2023, Kraków, Poland, volume 372, pages 1747–1754. IOS Press, 2023b.
- Gianluca Moro, Luca Ragazzi, and Lorenzo Valgimigli. Carburacy: Summarization models tuning and comparison in eco-sustainable regimes with a novel carbon-aware accuracy. In AAAI, pages 1–9. AAAI Press, 2023c.
- 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), 2023d. ISSN 1424-8220. doi: 10.3390/s23073542. URL <https://www.mdpi.com/1424-8220/23/7/3542>.
- Gianluca Moro, Luca Ragazzi, Lorenzo Valgimigli, and Lorenzo Molfetta. Retrieve-and-rank end-to-end summarization of biomedical studies. In SISAP 2023, A Coruña, Spain, October 9-11, 2023, Proceedings, volume 14289, pages 64–78. Springer, 2023e.
- Gianluca Moro, Luca Ragazzi, Lorenzo Valgimigli, et al. Efficient memory-enhanced transformer for long-document summarization in low-resource regimes. Sensors, 23(7):3542, 2023f.
- Gianluca Moro, Nicola Piscaglia, Luca Ragazzi, and Paolo Italiani. Multi-language transfer learning for low-resource legal case summarization. Artif. Intell. Law, 32(4):1111–1139, 2024a. doi: 10.1007/S10506-023-09373-8. URL <https://doi.org/10.1007/s10506-023-09373-8>.
- 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 11, 2024. OpenReview.net, 2024b.
- Gabriel Murray, Steve Renals, and Jean Carletta. Extractive summarization of meeting recordings. In INTERSPEECH, pages 593–596. ISCA, 2005.
- 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 EMNLP, pages 1797–1807, Brussels, Belgium, October-November 2018. ACL. doi: 10.18653/v1/D18-1206. URL <https://aclanthology.org/D18-1206>.
- Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. Scispacy: Fast and robust models for biomedical natural language processing. In BioNLP@ACL 2019, Florence, Italy, August 1, 2019, pages 319–327. ACL, 2019.
- 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. URL https://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf.
- Yixin Nie, Adina Williams, Emily Dinan, et al. Adversarial NLI: A new benchmark for natural language understanding. In *ACL 2020*, Online, July 5-10, 2020, pages 4885–4901. ACL, 2020.
- Joel Niklaus, Veton Matoshi, Pooja Rani, Andrea Galassi, Matthias Stürmer, and Ilias Chalkidis. LEX-TREME: 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. URL <https://aclanthology.org/2023.findings-emnlp.200>.
- Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*, 2019.
- John Edison Arevalo Ovalle, Tamar Solorio, Manuel Montes-y-Gómez, and Fabio A. González. Gated multimodal units for information fusion. In *Proc. of ICLR*. OpenReview.net, 2017. URL https://openreview.net/forum?id=S12_nqu0e.
- Yue Pan, Minxuan Zhou, Chonghan Lee, Zheyu Li, Rishika Kushwah, Vijaykrishnan Narayanan, and Tajaana Rosing. PRIMATE: processing in memory acceleration for dynamic token-pruning transformers. In *Proceedings of the 29th Asia and South Pacific Design Automation Conference, ASPDAC 2024*, Incheon, Korea, January 22-25, 2024, pages 557–563. IEEE, 2024a. doi: 10.1109/ASP-DAC58780.2024.10473968. URL <https://doi.org/10.1109/ASP-DAC58780.2024.10473968>.
- Zhuoshi Pan et al. Lmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 963–981. Association for Computational Linguistics, 2024b.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL <https://aclanthology.org/P02-1040>.
- Shantipriya Parida and Petr Motlíček. 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. doi: 10.18653/V1/D19-1616. URL <https://doi.org/10.18653/v1/D19-1616>.
- Pulkit Parikh, Harika Abburi, Niyati Chhaya, Manish Gupta, and Vasudeva Varma. Categorizing Sexism and Misogyny through Neural Approaches. *ACM Trans. on Web*, 15(4):1–31, 2021.
- Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 3967–3976. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00409. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Park_Relational_Knowledge_Distillation_CVPR_2019_paper.html.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, et al. 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, *NeurIPS*, pages 8024–8035, 2019.

- Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. In *ICLR (Poster)*. OpenReview.net, 2018.
- Felix Petersen, Hilde Kuehne, Christian Borgelt, and Oliver Deussen. Differentiable top-k classification learning. In *International Conference on Machine Learning*, pages 17656–17668. PMLR, 2022.
- Maxime Peyrard. A simple theoretical model of importance for summarization. In *ACL*, pages 1059–1073, Florence, Italy, July 2019. ACL. doi: 10.18653/v1/P19-1101. URL <https://aclanthology.org/P19-1101>.
- Jason Phang, Yao Zhao, and Peter J. Liu. Investigating efficiently extending transformers for long input summarization. In *EMNLP 2023, Singapore, December 6-10, 2023*, pages 3946–3961. ACL, 2023.
- Laura Plaza, Jorge Carrillo-de-Albornoz, Víctor Ruiz, Alba Maeso, Berta Chulvi, Paolo Rosso, Enrique Amigó, Julio Gonzalo, Roser Morante, and Damiano Spina. Overview of EXIST 2024 – Learning with Disagreement for Sexism Identification and Characterization in Social Networks and Memes (Extended Overview). In *Working Notes of CLEF*, 2024.
- Laura Plaza, Jorge Carrillo-de-Albornoz, Iván Árcos, Paolo Rosso, Damiano Spina, Enrique Amigó, Julio Gonzalo, and Roser Morante. EXIST 2025: Learning with Disagreement for Sexism Identification and Characterization in Tweets, Memes, and TikTok Videos. In *Proc. of the 47th ECIR*, volume 15576 of *Lecture Notes in Computer Science*, pages 442–449. Springer, 2025. doi: 10.1007/978-3-031-88720-8_65.
- Raul Puri, Ryan Spring, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Training question answering models from synthetic data. *arXiv preprint arXiv:2002.09599*, 2020.
- Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897, 2020. doi: 10.1007/s11431-020-1647-3.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019a.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Technical Report*, (1):1–24, 2019b.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.
- 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. *Journal of Machine Learning Research*, 21(140):1–67, 2020.
- Luca Ragazzi, Paolo Italiani, Gianluca Moro, and Mattia Panni. What are you token about? differentiable perturbed top-k token selection for scientific document summarization. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 9427–9440, 2024a.
- Luca Ragazzi, Paolo Italiani, Gianluca Moro, et al. What are you token about? differentiable perturbed top-k token selection for scientific document summarization. In *ACL*, August 11-16, pages 9427–9440. ACL, 2024b.
- Luca Ragazzi, Gianluca Moro, Stefano Guidi, and Giacomo Frisoni. Lawsuit: a large expert-written summarization dataset of italian constitutional court verdicts. *Artificial Intelligence and Law*, 2024c. URL <https://api.semanticscholar.org/CorpusID:272530787>.

- Luca Ragazzi, Gianluca Moro, Lorenzo Valgimigli, and Riccardo Fiorani. Cross-document distillation via graph-based summarization of extracted essential knowledge. *IEEE Transactions on Audio, Speech and Language Processing*, 33:518–527, 2025.
- Marc’Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *ICLR (Poster)*, 2016.
- 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. doi: 10.18653/V1/D19-1500. URL <https://doi.org/10.18653/v1/D19-1500>.
- Mohammad Zia Ur Rehman, Sufyaan Zahoor, Areeb Manzoor, Musharaf Maqbool, and Nagendra Kumar. A Context-Aware Attention and Graph Neural Network-based Multimodal Framework for Misogyny Detection. *Information Processing & Management*, 62(1):103895, 2025.
- Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. ACL, 2019a.
- Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019b.
- Leonardo Ribeiro, Mengwen Liu, Iryna Gurevych, Markus Dreyer, et al. Factgraph: Evaluating factuality in summarization with semantic graph representations. In *NAACL-HLT*, pages 3238–3253. ACL, 2022.
- Louise Richardson-Self. Woman-Hating: On Misogyny, Sexism, and Hate Speech. *Hypatia*, 33(2):256–272, 2018.
- Korbinian Riedhammer, Benoît Favre, and Dilek Hakkani-Tür. A keyphrase based approach to interactive meeting summarization. In *SLT*, pages 153–156. IEEE, 2008.
- Giulia Rizzi, Francesca Gasparini, Aurora Saibene, Paolo Rosso, and Elisabetta Fersini. Recognizing misogynous memes: Biased models and tricky archetypes. *Information Processing & Management*, 60(5):103474, 2023.
- Giulia Rizzi, David Gimeno-Gómez, Elisabetta Fersini, and Carlos-D Martínez-Hinarejos. PINK at EXIST2024: A Cross-Lingual and Multi-Modal Transformer Approach for Sexism Detection in Memes. *Working Notes of CLEF*, 2024.
- Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. Okapi at trec-3. *Nist Special Publication Sp*, 109:109, 1995.
- Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. *TACL*, 8:842–866, 2021.
- Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. 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/1412.6550>.
- Federico Ruggeri and Gaetano Signorelli. Interlocking-free selective rationalization through genetic-based learning. In *ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 1175–1191. ACL, 2025.

- Harvey Sacks, Emanuel A. Schegloff, and Gail Jefferson. A simplest systematics for the organization of turn taking for conversation. In *Studies in the Organization of Conversational Interaction*, pages 7–55. Elsevier, 1978. ISBN 978-0-12-623550-0. doi: <https://doi.org/10.1016/B978-0-12-623550-0.50008-2>.
- Ali Riahi Samani, Tianhao Wang, Kangshuo Li, and Feng Chen. Large Language Models with Reinforcement Learning from Human Feedback Approach for Enhancing Explainable Sexism Detection. In *Proc. of the 31st ICCL*, pages 6230–6243, 2025.
- Michael Eli Sander, Joan Puigcerver, Josip Djolonga, et al. Fast, differentiable and sparse top-k: a convex analysis perspective. In *ICML, Honolulu, Hawaii, USA*, volume 202, pages 29919–29936. PMLR, 2023.
- 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>.
- Timo Schick and Hinrich Schütze. Generating datasets with pretrained language models. *arXiv preprint arXiv:2104.07540*, 2021.
- John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, et al. High-dimensional continuous control using generalized advantage estimation. In *ICLR (Poster)*, 2016.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, et al. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.
- Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. Answers unite! unsupervised metrics for reinforced summarization models. In *EMNLP/IJCNLP (1)*, pages 3244–3254. ACL, 2019.
- Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *ACL (1)*, pages 1073–1083. ACL, 2017.
- Eva Sharma, Luyang Huang, Zhe Hu, and Lu Wang. An entity-driven framework for abstractive summarization. In *EMNLP/IJCNLP (1)*, pages 3278–3289. ACL, 2019.
- Chengchao Shen, Xinchao Wang, Jie Song, Li Sun, and Mingli Song. Amalgamating knowledge towards comprehensive 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 3068–3075. AAAI Press, 2019. doi: 10.1609/AAAI.V33I01.33013068. URL <https://doi.org/10.1609/aaai.v33i01.33013068>.
- 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. *Advances in Neural Information Processing Systems*, 35:13158–13173, 2022.
- Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2018. doi: 10.1016/j.physd.2019.132306.
- Koustuv Sinha, João Sedoc, and Samuel R. Bowman. Masked language modeling and beyond: Self-supervised learning for nlp. *Transactions of the Association for Computational Linguistics*, 9:727–742, 2021.
- 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.

- 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. doi: 10.1145/3462757.3466094. URL <https://doi.org/10.1145/3462757.3466094>.
- Vivek Srivastava, Savita Bhat, and Niranjan Pedanekar. A few good sentences: Content selection for abstractive text summarization. In Danai Koutra, Claudia Plant, Manuel Gomez Rodriguez, Elena Baralis, and Francesco Bonchi, editors, Machine Learning and Knowledge Discovery in Databases: Research Track - European Conference, ECML PKDD 2023, Turin, Italy, September 18-22, 2023, Proceedings, Part IV, volume 14172 of Lecture Notes in Computer Science, pages 124–141. Springer, 2023. doi: 10.1007/978-3-031-43421-1_8. URL https://doi.org/10.1007/978-3-031-43421-1_8.
- Statista. Most popular global mobile messaging apps 2022. <https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/>, 2022. Online; accessed 12 April 2023.
- Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. Dream: A challenge data set and models for dialogue-based reading comprehension. Transactions of the Association for Computational Linguistics, 7:217–231, 2019.
- Xiaofei Sun and Xiaoya Li et al. Text classification via large language models. In EMNLP 2023, Singapore, December 6-10, 2023, 2023.
- Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. Abstractive document summarization with a graph-based attentional neural model. In ACL (1), pages 1171–1181. ACL, 2017.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, et al. Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805, 2023.
- Gemma Team and Thomas et al. Mesnard. Gemma: Open models based on gemini research and technology. arXiv preprint arXiv:2403.08295, 2024.
- Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. Large language models can accurately predict searcher preferences. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 1930–1940, 2024.
- Lin Tian, Nannan Huang, and Xiuzhen Zhang. Efficient multilingual sexism detection via large language model cascades. In CLEF (Working Notes), pages 1083–1090, 2023.
- Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. CoRR, abs/1910.10699, 2019. URL <http://arxiv.org/abs/1910.10699>.
- Petter Törnberg. Chatgpt-4 outperforms experts and crowd workers in annotating political twitter messages with zero-shot learning. CoRR, abs/2304.06588, 2023.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, et al. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971, 2023a.
- 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>.
- Hugo Touvron, Louis Martin, Kevin Stone, et al. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023c.
- Hai-Long Trieu, Thy Thy Tran, Anh-Khoa Duong Nguyen, Anh Nguyen, et al. Deepeventmine: end-to-end neural nested event extraction from biomedical texts. *Bioinform.*, 36(19):4910–4917, 2020.
- Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*, 2023.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017a.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need. In *NeurIPS 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017b.
- Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, et al. Graph attention networks. In *ICLR (Poster)*. OpenReview.net, 2018.
- Joseph B. Walther. Social Media and Online Hate. *Current Opinion in Psychology*, 45:101298, 2022. ISSN 2352-250X.
- Xiaojun Wan. An exploration of document impact on graph-based multi-document summarization. In *EMNLP*, pages 755–762. ACL, 2008.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Ting Ji, Xia Zhang, and et al. Decoder-only language models are strong abstractive summarizers. *arXiv preprint arXiv:2306.01245*, 2023a.
- Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, et al. Heterogeneous graph neural networks for extractive document summarization. In *ACL*, pages 6209–6219. ACL, 2020a.
- Jiaan Wang, Yunlong Liang, Fandong Meng, Haoxiang Shi, et al. Is chatgpt a good NLG evaluator? A preliminary study. *CoRR*, abs/2303.04048, 2023b. doi: 10.48550/arXiv.2303.04048. URL <https://doi.org/10.48550/arXiv.2303.04048>.
- 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*, 2020b.
- Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based LSTM for aspect-level sentiment classification. In *EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 606–615. ACL, 2016.
- 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*, 2022.

- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, et al. Transformers: State-of-the-art natural language processing. In *EMNLP*, pages 38–45, Online, October 2020. ACL. doi: 10.18653/v1/2020.emnlp-demos.6. URL <https://aclanthology.org/2020.emnlp-demos.6>.
- Chien-Sheng Wu, Steven C.H. Hoi, Richard Socher, and Caiming Xiong. TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue. In *EMNLP*, pages 917–929, Online, November 2020. ACL. doi: 10.18653/v1/2020.emnlp-main.66. URL <https://aclanthology.org/2020.emnlp-main.66>.
- Chien-Sheng Wu, Linqing Liu, Wenhao Liu, Pontus Stenetorp, et al. Controllable abstractive dialogue summarization with sketch supervision. In *ACL-IJCNLP 2021*, pages 5108–5122, Online, August 2021a. ACL. doi: 10.18653/v1/2021.findings-acl.454. URL <https://aclanthology.org/2021.findings-acl.454>.
- Lingfei Wu, Yu Chen, Kai Shen, Xiaojie Guo, et al. Graph neural networks for natural language processing: A survey. *CoRR*, abs/2106.06090, 2021b.
- Wenhao Wu, Wei Li, Xinyan Xiao, Jiachen Liu, et al. BASS: boosting abstractive summarization with unified semantic graph. In *ACL/IJCNLP (1)*, pages 6052–6067. ACL, 2021c.
- Bo Xu, Erchen Yu, Jiahui Zhou, Hongfei Lin, and Linlin Zong. HyperHatePrompt: A Hypergraph-based Prompting Fusion Model for Multimodal Hate Detection. In *Proc. of the 31st ICCL*, pages 3825–3835. ACL, 2025.
- Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *Advances in neural information processing systems*, 35:38571–38584, 2022.
- Alok Yadav and Shubham Shukla. A survey on abstractive text summarization using deep learning. *Expert Systems with Applications*, 184:115477, 2021. doi: 10.1016/j.eswa.2021.115477.
- Tao Yang, Fei Ma, Xiaoling Li, et al. Dtatrans: Leveraging dynamic token-based quantization with accuracy compensation mechanism for efficient transformer architecture. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42(2):509–520, 2022.
- Tao Yang, Fei Ma, Xiaoling Li, et al. Dtatrans: Leveraging dynamic token-based quantization with accuracy compensation mechanism for efficient transformer architecture. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 42(2):509–520, 2023.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.
- Zichao Yang, Diyi Yang, Chris Dyer, et al. Hierarchical attention networks for document classification. In *NAACL*, pages 1480–1489, 2016.
- Deming Ye, Yankai Lin, Yufei Huang, and Maosong Sun. TR-BERT: dynamic token reduction for accelerating BERT inference. In *NAACL-HLT 2021*, Online, June 6-11, 2021, pages 5798–5809. ACL, 2021a.
- Deming Ye, Yankai Lin, Yufei Huang, and Maosong Sun. Tr-bert: Dynamic token reduction for accelerating bert inference. *arXiv preprint arXiv:2105.11618*, 2021b.
- 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. doi: 10.18653/V1/2022.EMNLP-MAIN.801. URL <https://doi.org/10.18653/v1/2022.emnlp-main.801>.

- 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. arXiv preprint arXiv:2202.07922, 2022.
- Junho Yim, Donggyu Joo, Ji-Hoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, pages 7130–7138. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.754. URL <https://doi.org/10.1109/CVPR.2017.754>.
- Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyeong Park. Gpt3mix: Leveraging large-scale language models for text augmentation. arXiv preprint arXiv:2104.08826, 2021.
- Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, et al. A survey of knowledge-enhanced text generation. CoRR, abs/2010.04389, 2020.
- 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. arXiv preprint arXiv:2306.15895, 2023.
- Weizhe Yuan, Graham Neubig, and Pengfei Liu. Bartscore: Evaluating generated text as text generation. In NeurIPS 2021, December 6-14, 2021, virtual, pages 27263–27277, 2021a.
- 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, NeurIPS, pages 27263–27277, 2021b. URL <https://proceedings.neurips.cc/paper/2021/hash/e4d2b6e6fdeca3e60e0f1a62fee3d9dd-Abstract.html>.
- Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. 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=Sks9_ajex.
- Farooq Zaman, Matthew Shardlow, Saeed-Ul Hassan, et al. HTSS: A novel hybrid text summarisation and simplification architecture. Inf. Process. Manag., 57(6):102351, 2020.
- Klaus Zechner. Automatic summarization of open-domain multiparty dialogues in diverse genres. Comput. Linguistics, 28(4):447–485, 2002.
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In ICML 2020, 13-18 July 2020, Virtual Event, volume 119, pages 11328–11339. PMLR, 2020a.
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In ICML, volume 119 of Proceedings of Machine Learning Research, pages 11328–11339. PMLR, 2020b.
- Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. Graph Convolutional Networks: A Comprehensive Review. Computational Social Networks, 6(1):1–23, 2019.
- Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, et al. OPT: open pre-trained transformer language models. CoRR, abs/2205.01068, 2022. doi: 10.48550/arXiv.2205.01068. URL <https://doi.org/10.48550/arXiv.2205.01068>.
- Tianyi Zhang, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In International Conference on Learning Representations, 2020c. URL <https://openreview.net/forum?id=SkeHuCVFDr>.

- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, et al. Bertscore: Evaluating text generation with BERT. In ICLR. OpenReview.net, 2020d.
- Weiqi Zhang, Hechuan Shen, Tianyi Lei, Qian Wang, Dezhong Peng, and Xu Wang. GLQA: A generation-based method for legal question answering. In International Joint Conference on Neural Networks, IJCNN 2023, Gold Coast, Australia, June 18-23, 2023, pages 1–8. IEEE. doi: 10.1109/IJCNN54540.2023.10191483. URL <https://doi.org/10.1109/IJCNN54540.2023.10191483>.
- Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. NeurIPS, 28, 2015.
- Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4320–4328, 2018.
- Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, et al. Adversarial feature matching for text generation. In Doina Precup and Yee Whye Teh, editors, ICML, volume 70 of Proceedings of Machine Learning Research, pages 4006–4015. PMLR, 2017. URL <http://proceedings.mlr.press/v70/zhang17b.html>.
- Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, et al. DIALOGPT : Large-scale generative pre-training for conversational response generation. In ACL, pages 270–278, Online, July 2020e. ACL. doi: 10.18653/v1/2020.acl-demos.30. URL <https://aclanthology.org/2020.acl-demos.30>.
- Lulu Zhao, Weiran Xu, and Jun Guo. Improving abstractive dialogue summarization with graph structures and topic words. In COLING, pages 437–449, Barcelona, Spain (Online), December 2020. COLING. doi: 10.18653/v1/2020.coling-main.39. URL <https://aclanthology.org/2020.coling-main.39>.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, et al. 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. doi: 10.1145/3462757.3466088. URL <https://doi.org/10.1145/3462757.3466088>.
- Ming Zhong, Yang Liu, Yichong Xu, Chenguang Zhu, et al. Dialoglm: Pre-trained model for long dialogue understanding and summarization. In AAAI, pages 11765–11773. AAAI Press, 2022a. URL <https://ojs.aaai.org/index.php/AAAI/article/view/21432>.
- Ming Zhong, Yang Liu, Yichong Xu, Chenguang Zhu, et al. Dialoglm: Pre-trained model for long dialogue understanding and summarization. In AAAI, pages 11765–11773. AAAI Press, 2022b.
- Chunting Zhou, Graham Neubig, and Luke Zettlemoyer. Trends in natural language understanding and generation: A deep learning perspective. ACM Computing Surveys, 55(11):1–38, 2023a. doi: 10.1145/3519371.
- 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, 2023b. doi: 10.48550/ARXIV.2308.03279. URL <https://doi.org/10.48550/arXiv.2308.03279>.
- Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, and Jiantao Jiao. Starling-7b: Improving llm helpfulness & harmlessness with rlai, November 2023.

- Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, et al. Enhancing factual consistency of abstractive summarization. In NAACL-HLT, pages 718–733. ACL, 2021.
- Tong Zhu, Leida Li, Jufeng Yang, Sicheng Zhao, and Xiao Xiao. Multimodal Emotion Classification with Multi-Level Semantic Reasoning Network. IEEE Trans. on Multimedia, 25:6868–6880, 2022.
- Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, et al. Fine-tuning language models from human preferences. CoRR, abs/1909.08593, 2019.



Finanziato dall'Unione europea- Next Generation EU, Missione 4, Componente 2, Investimento 3.3 (D.M. 117/2023) CUP J33C22001400009