Alma Mater Studiorum – Università di Bologna
In collaborazione con LAST-JD consortium:
Università degli studi di Torino
Universitat Autonoma de Barcelona
Mykolas Romeris University
Tilburg University
e in cotutela con
THE Luxembourg University

DOTTORATO DI RICERCA IN

Erasmus Mundus Joint International Doctoral Degree in
Law, Science and Technology

Ciclo 30^{o}

Settore Concorsuale: INF/01
Settore Scientifico Disciplinare: 01/B1

TITOLO TESI

Multimodal Legal Information Retrieval

Presentata da: Kolawole John ADEBAYO

Coordinatore Dottorato
Prof. Giovanni Sartor

Supervisore
Prof. Guido Boella
Dr. Luigi Di Caro

Esame finale anno 2018

Alma Mater Studiorum – Università di Bologna
in partnership with LAST-JD Consoritum
Università degli studi di Torino
Universitat Autonoma de Barcelona
Mykolas Romeris University
Tilburg University
and in cotutorship with the
THE University of Luxembourgh

PhD Programme in

Erasmus Mundus Joint International Doctoral Degree in
Law, Science and Technology

Cycle 30^{o}

Settore Concorsuale di afferenza: INF/01
Settore Scientifico disciplinare: 01/B1

Title of the Thesis

Multimodal Legal Information Retrieval

Submitted by: Kolawole John ADEBAYO

The PhD Programme Coordinator Prof. Giovanni Sartor	Supervisor (s) Prof. Guido Boella Dr. Luigi Di Caro

Year 2018

PhD-FSTC-2018-03 The Faculty of Sciences, Technology and Communication

University of Bologna
Law School

DISSERTATION

Defence held on 27/03/2018 in Bologna
to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG EN INFORMATIQUE

AND

DOTTORE DI RICERCA
in Law, Science and Technology

By

ADEBAYO, KOLAWOLE JOHN
Born on 31st January, 1986 in Oyo (Nigeria).

Multimodal Legal Information Retrieval

Dissertation Defence Committee

Prof. Marie-Francine Moens, Chairman
Katholieke Universiteit, Belgium
Prof. Henry Prakken, Vice-Chairman
Universiteit Utrecht, Netherlands
Prof. Schweighofer Erich, Member
University of Vienna, Vienna
Prof. Leon van der Torre, Dissertation Supervisor
Université du Luxembourg, Luxembourg
Prof. Guido Boella, Dissertation Supervisor
Università degli Studi di Torino, Italy
Prof. Monica Palmirani, Discussant
Università di Bologna, Italy
Prof. Luigi Di Caro, Discussant
Università degli Studi di Torino, Italy

Declaration of Authorship

I, Kolawole John ADEBAYO, declare that this thesis titled, "Multimodal Legal Information Retrieval" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"The limits of my language are the limits of my world."

Ludwig Wittgenstein

Abstract

Kolawole John ADEBAYO

Multimodal Legal Information Retrieval

The goal of this thesis is to present a multifaceted way of inducing semantic representation from legal documents as well as accessing information in a precise and timely manner. The thesis explored approaches for semantic information retrieval (IR) in the Legal context with a technique that maps specific parts of a text to the relevant concept. This technique relies on text segments, using the Latent Dirichlet Allocation (LDA), a topic modeling algorithm for performing text segmentation, expanding the concept using some Natural Language Processing techniques, and then associating the text segments to the concepts using a semi-supervised Text Similarity technique. This solves two problems, i.e., that of user specificity in formulating query, and information overload, for querying a large document collection with a set of concepts is more fine-grained since specific information, rather than full documents is retrieved. The second part of the thesis describes our Neural Network Relevance Model for *E-Discovery* Information Retrieval. Our algorithm is essentially a feature-rich *Ensemble* system with different component Neural Networks extracting different relevance signal. This model has been trained and evaluated on the TREC Legal track *2010* data. The performance of our models across board proves that it capture the semantics and relatedness between query and document which is important to the Legal Information Retrieval domain.

Subject: Legal Informatics.

Keywords: Convolutional Neural Network, Concept, Concept-based IR, CNN, E-Discovery, Eurovoc, EurLex, Information Retrieval, Document Retrieval, Legal Information Retrieval, Semantic Annotation, Semantic Similarity, Latent Dirichlet Allocation, LDA, Long Short-Term Memory, LSTM, Natural Language Processing, Neural Information Retrieval Neural Networks, Text Segmentation, Topic Modeling

Il Riassunto

L'obiettivo di questa tesi è quello di presentare un modo sfaccettato di accesso alle informazioni da un curpus di documenti legali in modo preciso ed efficiente. Il lavoro inizia con un'esplorazione degli approcci relativi al recupero di informazioni semantiche (Information Retrieval o IR) nel contesto giuridico con una tecnica che mappa alcune parti di un testo a specifici concetti di una ontologia, basandosi su una segmentazione semantica dei testi. Tecniche di elaborazione del linguaggio naturale vengono poi utilizzate per associare i segmenti di testo ai concetti utilizzando una tecnica di similarità testuale. Pertanto, interrogando un documento legale di grandi dimensioni con una serie di concetti è possibile recuperare segmenti di testo ad una grana più fine, piuttosto che i documenti completi originari. La tesi si conclude con la descrizione di un classificatore di reti neurali per l'E-Discovery. Questo modello è stato addestrato e valutato sui dati della legal track TREC 2010, ottenendo una performance in grado di dimostrare che le più recenti tecniche di neural computing possono fornire buone soluzioni al recupero di informazioni che vanno dalla gestione dei documenti, di informazioni aziendali e di scenario relativi all'E-Discovery.

Oggetto: Informatica legale.

Parole chiave: Recupero di informazioni, Annotazione semantica, Somiglianza semantica, Allineamento testuale, Risposte automatiche a domande legali, Reti neurali, Eurlex, Eurovoc, Estrazione di keyphrase.

Acknowledgements

I would like to thank the almighty God, the giver of life and the one in whom absolute power and grace resides. Many people have made the completion of this doctoral programme a reality. My thanks go to Prof. Monica Palmirani, the coordinator of the LAST-JD programme, and other academic committee members for finding me suitable for this doctoral programme.

I thank my supervisors, Prof. Guido Boella, Prof. Leon Van Der Torre, and Dr. Luigi Di Caro for their guidance, advise, countless periods of discussion, and for putting up with my numerous deadline-day paper review request. Guido has been a father figure, and I could not have asked for more. Leon gave me tremendous support throughout the Ph.D. journey. I have participated at many conferences partly due to him. Luigi, a big part of the compliment goes to you! You never stopped reminding me that I could do better.

I thank my wife, Oluwatumininu, and beautiful daughters -Joyce and Hillary for the love and understanding even when I am miles and months away. This thesis is only possible because of you! I thank my mum, Modupe, and siblings -Olaide, Adeboyin, Adefemi, Abiola and Oluwatobi for their love and support at all time. I do not forget other family members whom I cannot mention for the sake of space. God keep and bless you all.

I thank the friends and colleagues that I have met during the period of the doctoral research, You all have made my stay in the many countries where I have worked memorable. Marc Beninati, thanks for being a friend and brother. To Livio Robaldo, thanks for your encouragement, you gave me fire when I needed it the most!,Dina Ferrari, thanks for your usual support. To Antonia -my Landlady in Barcelona, and Prof. Roig -my host, you made Barcelona to be eternally engraved in my heart.

Lastly, I thank the European Commission without whose scholarship I would not have been a part of this prestigious academic programme.

Contents

List of Figures

List of Tables

Part I

General Introduction

Chapter 1

INTRODUCTION

1.1 Introduction

The early days of this research were pre-occupied with an endless search for relevant publications, books, and other interesting information that is useful to the research topic. One of my favourite generic search keyphrases was *'Handbook of Legal Information Retrieval pdf'*. At least, I wanted a detailed but specific book(s) on Legal Information research in PDF format. One of the results of the Google hit is a book titled -'*Handbook of Legal Procedures of Computer and Network*'. To the uninformed mind, this would be relevant, more so, both the query and the retrieved item have a few things in common, at least lexically, i.e., the retrieved document matches the query words 'Handbook' and Legal'. However, it turns out that the Google search engine got the search intent wrong, and conversely, the retrieved article reflects that. The experience was an eye-opener to the practical realities of the complexity of information retrieval task.

Simply put, the operation that I carried out is generally referred to as Information Retrieval, from web search to searching for a document on one's computer -it is what countless of people do daily, and the goal of every search activity is to determine the presence or absence of an information of specific interest to the user, often from a mass of information that is available (Salton and McGill, 1986). The information that is of specific interest to the user is often called the *information need*, and following my analogy, the set of search words that I used in presenting my information need is what is generally called the *query* (Baeza-Yates and Ribeiro-Neto, 1999).

Information Retrieval (IR) is not a new discipline, however, the content that is being searched (i.e., paper document, electronic document, musical files etc.), the approaches and methods employed for search, as well as the technology involved, has evolved over time. For instance, many decades ago, information was mostly available in written or printed forms. The task of organizing documents was both trivial and difficult. It was trivial to the extent that the job was to stack paper files on top of one another, and orderly arrange the stack inside a cabinet. However, it was difficult to the extent that the retrieval of any file requires sorting and checking through thousands or millions of files. Librarians, in the past, are the first set of people who happen to deal with a lot of books and paper documents, and thus, they introduced some indexing mechanism as a way of simplifying their task and reducing the response time of library patrons. Books by nature have specific meta-data by which they can be organized. The meta-data includes the name of author(s), subject, title of the book, and any other bibliographic categories. A search can then be made using the meta-data to retrieve the needed book. Interestingly, the system worked quite well because most of the documents are somewhat structured. To be specific, we say that a document that has some attributes, e.g. meta-data through which it can be identified is *semi-structured* while the ones with a well-linked information which has been organized into different records according to a well-defined syntax in a database are said to be *structured*. A free text is classified as being unstructured. Generally, an overwhelming amount of documents have no structure, and in such cases, it would be difficult to index the documents since there exist no meta-data attributes. Despite the impressive effort of Researchers like Palmirani et. al., (Palmirani and Vitali, 2011; Palmirani and Vitali, 2012) in introducing XML standard like the *Akoma Ntoso*, for formatting juridical documents, a commanding percentage of legal documents are still unstructured (Zeleznikow, 2005). Generally, when we talk about a document in this thesis, unless otherwise specified, we refer to an unstructured textual document, for it is the most important for the solutions presented in this thesis.

With the advent of computers, the earliest forms of search uses keywords in case of a free-text search or the more sophisticated structured database solution. The latter, i.e., the structured database provides a parallelized search by linking information that is segmented into different tables, based on some unifying attributes, and a search conveying user's need is carried out with the aid of a query language called the *structured query language* (SQL). The former, i.e., the keyword search, offers an overlapping solution for a free-text search based on some explicit keywords that appear frequently in the body of the documents to be retrieved. However, keyword search has become inefficient owing to the *data explosion* and also due to a number of language variability issues such as Synonymy and Polysemy (Croft, Metzler, and Strohman, 2010).

1.2 Legal Information Retrieval

A field that the computer keeps revolutionizing is the Legal field; bringing about an uncommon trend and evolution in both the practice of law and the attitudes and skills of the practitioners. For instance, the automation of legal processes has prompted lawyers, paralegals, legal secretaries and other legal professionals become proficient in an ever-increasing array of word processing, spreadsheet, telecommunications, database, presentation, courtroom operation and document management, evidential innovations and legal research software. The increasing use of computer, coupled with the growth of the internet, the adaptation by practitioners of Information and Communication Technology (ICT) tools and the emerging powerful database technologies implies that data accumulates in an unprecedented manner, and readily in electronic form, and at a proportion than any practitioner can contend with(Baron, 2011). The deluge of electronically stored information (ESI) has therefore practically necessitated developing frameworks for intelligent document processing and extraction of useful knowledge needed for categorizing information, retrieving relevant information as well as other useful tasks. This is important since ESI is mostly unstructured (Oard et al., 2010).

Legal Information Retrieval (LIR) has for some years been the focus of research within the broader Artificial Intelligence and Law (AI & Law) field (Bench-Capon et al., 2012). The goal of LIR is to model the information-seeking behavior which lawyers exhibit when using a range of existing legal resources to find the information required for their work (Leckie, Pettigrew, and Sylvain, 1996). Van Opijnen and Santos (Van Opijnen and Santos, 2017) opined that information retrieval in the legal domain is not only quantitative (in terms of the amount of data to deal with), but also qualitative. For instance, the duties of a lawyer include research, drafting, negotiation, counseling, managing and argumentation, and therefore, an ideal LIR system should transcend the quantitative aspect like document retrieval only, but should explicitly model the complexities of the law and of legal information seeking behaviour. In addition, such a system must take cognizance of the distinguishing peculiarities of legal information, which aside its huge volume include document size, structure, heterogeneity of document types, self-contained documents, legal hierarchy, temporal aspects, the importance of citations etc. (Van Opijnen and Santos, 2017).

Legal practitioners are not unfamiliar with IR. Lawyers for years have had to reinforce their arguments by researching and quoting pre-existing court decisions. LexisNexis and Westlaw for instance, are popular commercial legal research service providers, who offer legal, regulatory and business information and analytics that help practitioners make more informed decisions, increase productivity and serve their clients better. Lexis, for example, offers search over a repository of United States (US) state and federal published case opinion, statutes, and laws etc. In addition, these companies provide other value-added services e.g., Westlaw's KeyCite system, which keeps track of the number of time, and the incidence for which the case was cited. However, these systems heavily rely on the Boolean retrieval model, which is also the prevalent approach for most Electronic Discovery (E-Discovery) systems.

The Boolean retrieval model is considered to have worked well for precedence search systems primarily because of the quality of the Boolean queries. For example, with the proximity-operator constraint, a user may specify that some particular terms in a document must be placed within a certain number of words or pages of each other. As we will see later in Chapter 2, the objective of Precedence search is different from that of E-Discovery. Since E-Discovery is a kind of ad-hoc search, in the remainder of this thesis, we interchangeably refer to it as ad-hoc search. While the goal of an Enterprise search is to have high precision, i.e., to retrieve few documents, taking into consideration relevance criterion such as an explicit temporal factor in order to determine the current binding precedent, ad-hoc search, on the other hand, focuses on achieving high recall, i.e., many documents that are ranked in their order of relevance. Almquist (Almquist, 2011) opined that these differences can be explained by the different roles that evidence and arguments play in legal proceedings.

Managing huge amount of ESI is not an easy task. The challenge is that of effective management, such that documents are organized in a way that an easy IR, extraction, searching, and indexing can be done. As in every other field, the Legal domain has also witnessed a boom in the amount of ESI produced e.g., in the law courts, government assemblies etc. This comes with the responsibility of developing retrieval techniques that scale up to this data and afford users the possibility of getting access to needed information in a timely and efficient manner.

1.3 Motivation

As technology becomes available to more people, especially in solving the day-to-day tasks, there continues to be a surge in the amount of unstructured text being produced. The legal community is not isolated in this regard because legal practitioners are constantly inundated with a huge amount of documents that have to be processed. Specifically, it is possible to break down the kind of search that is peculiar to the legal domain, and two important categories readily come to the mind, i.e., 1) Ad-hoc search which is technically what E-Discovery is all about, or 2) Enterprise search which is performed when search tools like *Lexis* and *Westlaw* are used by lawyers to search for information within a repository, or even when a document is being searched on a web database like the EUR-Lex or CELEX. The work in this thesis describes our novel ensemble NN model for the former, a semantic annotation-based system for the latter, as well as our approach for inducing deep semantic representation across a range of legal corpora and its application in Question Answering for the Legal domain.

With respect to the first case of the ad-hoc search, E-Discovery has over the years grown geometrically into a multi-billion dollar business and as shown in Figure 1.1, the software and services market is projected to eclipse $*16* billion by *2021*. It is expected that E-Discovery solutions and services will further empower organizations to streamline their business processes by providing the possibilities for obtaining, securing, searching, and processing electronic data effectively and efficiently. Furthermore, E-Discovery solutions and services have its tentacles spread across government, legal sector, Banking, Financial Services, and Insurance, healthcare, Telecom, Energy, Hospitality, Transportation, Entertainment, and education sector to mention a few. The major forces driving this market include focus on decreasing operational budget of legal departments, global increase in litigations, stringent compliance with policies and regulations worldwide, increase in mobile device penetration and usage.
In the United States of America (US), *19* million and *303,000* civil cases are filed in state and federal courts respectively each year, all with a total annual cost between $*200* - $*250*

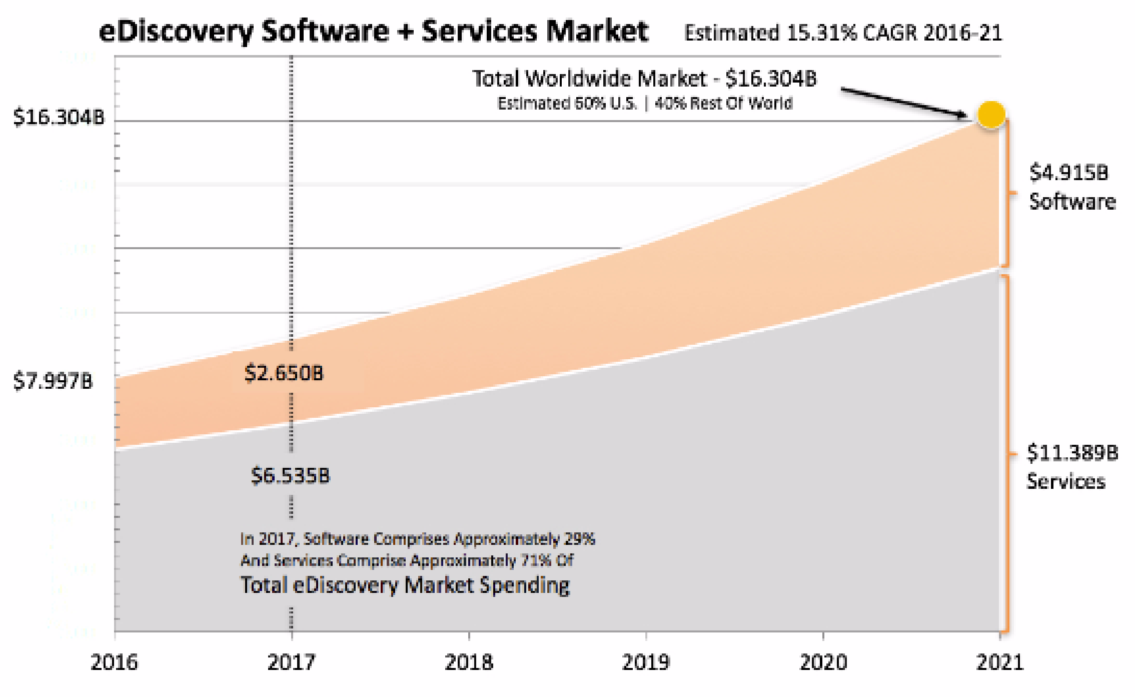


FIGURE 1.1: E-Discovery Software and Services Market Projection: 2016 – 2021. (Source: www.complexdiscovery.com)

billion. Of these filings, about *12.7* million cases involve contracts or torts. It is estimated that about *60%* of all civil cases involve discovery, and more importantly, about *20* to *50* percent of all costs in federal civil litigation are incurred to perform discovery, without including soft costs like business interruption. Putting all figures together, *discovery* is said to cost the United States an average of $*42.1* billion per year. To put this in proper perspective, if US E-Discovery was its own economic nation, it would rank *90*th out of *189* countries in the world[1].

However, as significant, important, and costly this process is, existing techniques for developing E-Discovery systems are based on the conventional information retrieval models, i.e., the Boolean model, Vector Space model, and Topic model (Wei, 2007; Oard et al., 2010; Pohl, 2012; Oard and Webber, 2013; Ayetiran, 2017). In practice, a manual approach could be used for review when collection size is small, however, Machine Learning (ML) approaches can be used to automatically reduce the size of the search space. Consequently, determining the relevance of a document can be viewed as a classification task, i.e., a document can either be relevant given a topic or query, otherwise the document is not relevant. Predictive coding techniques which use powerful ML classifiers have been proposed (Almquist, 2011; Hyman, 2012; Cormack and Grossman, 2014).

An exciting field of artificial intelligence in Computer science is the ML. A resurfaced branch of ML is the design of Neural Network (NN) systems, which when configured to have many layers are referred to as Deep Learning Neural Networks (DNN) (LeCun, Bengio, and Hinton, 2015). Given the recent success and state of the art performance of

[1] Statistics quoted herein were obtained from: http://blog.logikcull.com/estimating-the-total-cost-of-u-s-ediscovery

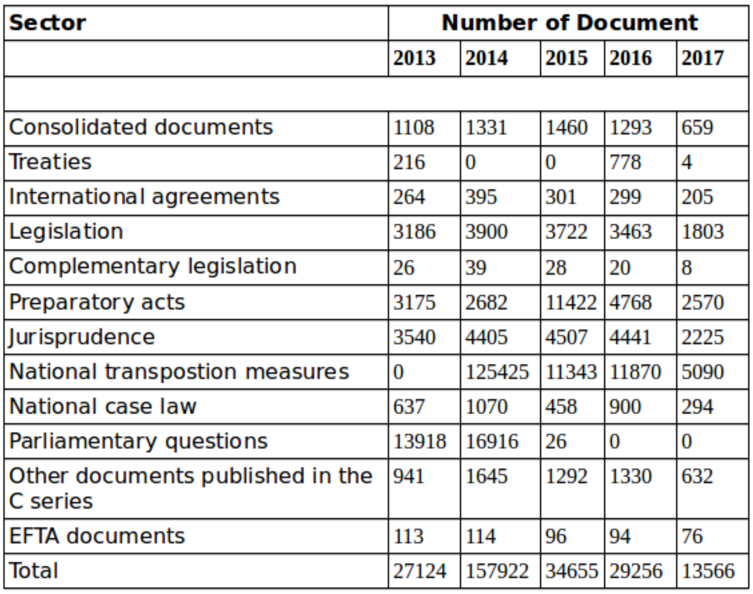
DNNs in several Natural Language Processing (NLP) tasks such as Machine Translation (Bahdanau, Cho, and Bengio, 2014; Cho et al., 2014; Sutskever, Vinyals, and Le, 2014), Image Recognition (Simonyan and Zisserman, 2014; He et al., 2016), and Speech Recognition (Hinton et al., 2012; Dahl et al., 2012). One of the goals of this thesis is to develop a DNN-based classifier for the E-Discovery task as a form of technology-assisted review. Moreover, when efficient technology-assisted review systems are deployed, they could help reduce risk, while also helping to drastically reduce the duration and budget of any review exercise.

As regards the Enterprise search, more than ever, the field of Law is generating information than anyone could have imagined years ago. This is unsurprising because the number of cases that are being tried in court keeps increasing. Also, there is an exponential growth in the amount of ESI produced both in the courts and government parliaments, especially with the crave for e-government and open-government (Baron, 2011). As an example, *EUR-Lex*[2] is a repository of legal documents from the European Union parliament and Table 1.2 shows the number of English documents that were added to the repository between 2013-2017. This huge volume of documents requires an effective and intelligent retrieval process.

A basic legal principle in many countries, especially where common law is practiced is *Stare Decisis*, which in a lay man's language means *decision governs*. The principle upholds the norms of legal precedent, i.e., past court cases are used as the standard for delivering future decisions. Because of this, old court cases are as relevant and important to lawyers as new court cases, hence, any case law search would require a scrutiny of every available case laws (no matter how old) in the repository. The problems of synonymy and polysemy, among other language variability issues have shown that the future of IR lies in understanding the meaning of a document's content. Perhaps, such meanings can be mapped to the relevant user intent. It is therefore important that a developed system should be able to provide seamless semantic-based retrieval even on a huge repository of several millions of old and new court cases. There are many desiderata for such a seamless semantic-based retrieval system (see section (2.3) for details), i.e., such a system should:

- Be robust to the different ways a user could present his/her information need (query).
- Transcend matching or retrieving based on words but rather based on the overall semantic/meaning of the intended document.
- Be able to retrieve specific portion (passage) of the document that may be of interest to the user.

[2] EUR-Lex is a collection EU governments data as well as data from national governments of EU countries. Entries cover treaties, international agreements, legislation, national case-law, preparatory acts, parliamentary questions etc. EUR-Lex is available at http://eur-lex.europa.eu/homepage.html



Sector	Number of Document				
	2013	2014	2015	2016	2017
Consolidated documents	1108	1331	1460	1293	659
Treaties	216	0	0	778	4
International agreements	264	395	301	299	205
Legislation	3186	3900	3722	3463	1803
Complementary legislation	26	39	28	20	8
Preparatory acts	3175	2682	11422	4768	2570
Jurisprudence	3540	4405	4507	4441	2225
National transpostion measures	0	125425	11343	11870	5090
National case law	637	1070	458	900	294
Parliamentary questions	13918	16916	26	0	0
Other documents published in the C series	941	1645	1292	1330	632
EFTA documents	113	114	96	94	76
Total	27124	157922	34655	29256	13566

FIGURE 1.2: EUR-Lex Content Statistics.
(Source: `http://eur-lex.europa.eu/statistics`)

Part II of this thesis describes a system that incorporates these desiderata. This part of our work makes use of documents from EUR-Lex, a web-based multilingual repository for European Union legislative documents. These documents are already manually labeled with concepts from Eurovoc[3], therefore allowing users to search for relevant documents from the repository by using the concepts as the query. The proposed system uses a pool of NLP techniques to aggregate and map the meaning of the user intent (i.e., concept) to the relevant parts of a text. This part of our work is referred to as Concept-based information retrieval.

Overall, the thesis adopts a structured approach to Legal Information Retrieval. Rather than fixating on a single case of information retrieval task, we developed different approaches for inducing semantic representation from legal text, and proposing approaches by which the induced representation can be algorithmically used in providing relevant, meaningful and useful information to users, at different levels of granularity. The techniques also rely on different legal corpora, tool chains, as well as algorithms.

1.4 Problem Statement

It is said that the primary challenge for lawyers, who unlike many other professionals live in the world of investigations and litigations in this age of exponential information explosion, is to devise a way to reasonably manage the Electronically Stored Information (ESI) by relying on the modern-day techniques (Baron, 2011). Civil discovery is a particular task that involves analysis and retrieval of relevant documents from a voluminous set

[3] Eurovoc is a taxonomy of concepts that describe some legal terms. It is available at http://eurovoc.europa.eu/.

of data. As an example, the author (Baron, 2011) cited an example of an examiner who had to review some *350* billion pages (*3* Peta-bytes) worth of data in a single discovery exercise.

Civil discovery obliges parties to a lawsuit to provide responsive documents that are sensitive to the case to each other provided that the request is not subject to a claim of privilege (Oard and Webber, 2013). Since most documents are now available as ESI, the term E-Discovery is often used. E-Discovery refers to the process by which one party (e.g., the plaintiff) is entitled to request evidence in ESI format, that is held by another party (e.g., the defendant) and that is relevant to some matter that is the subject of civil litigation (i.e., what is commonly called a "lawsuit"). This procedure, among many other challenging tasks for legal practitioners, often appears cumbersome with a high cost of the undertaking.

Three key problems affecting LIR have been identified in this study. The first is the problem of *user specificity*, i.e., how is the *information need represented* or *presented* to the system? The second problem is the notion of *relevance*. How do we determine what is relevant or what is not relevant, based on the specified user request?. Also, what constitutes relevance? Opijnen and Santos (Van Opijnen and Santos, 2017) give six dimensions of relevance that are of interest to LIR. The most important of this is the semantic relevance which is addressed in this thesis (*see section* 2.5). The preceding problems are intertwined. Ideally, a retrieval system typically assumes that a user fully understands his needs and able to feedback those needs into his thought process when constructing the query. However, Legal Information Systems (LIS) are mostly based on keywords, and by extension, the bag-of-words based Boolean models (Salton, 1971; Salton, Wong, and Yang, 1975; Salton, Fox, and Wu, 1983; Manning, Raghavan, and Schutze, 2008), which unfortunately do not fully capture the thought that a user has when formulating the query words. A Concept, being an abstraction of a general idea which may otherwise be described in detail with words, may be used to represent the user intent such that the user does not have to worry about how to specify the query.

Generally, BOW-based approaches which rely on word frequency have issues with polysemous words and synonyms. *Polysemy* is a term used for words which have multiple meanings for the same lexical or orthographic form, while *Synonymy* is a term that describes a word which has other words with exactly, closely related or substitutable meaning. Counting word frequency, couples with the two phenomenons highlighted above introduce some arbitrariness into how relevance is perceived by retrieval systems. In other words, both Polysemy and Synonymy impact the performance of a retrieval system negatively, and in different ways. For instance, while Synonymy degrades the *recall*, Polysemy degrades the *precision* of the system. Nevertheless, their eventual effect on an IR system is called the *Query-Document Mismatch*. We say that a Query-Document mismatch occurs when the query representation and the document representation expresses the same concept but the IR system is unable to realize the relatedness, hence, omitting the document as though it is not relevant to the query. Researchers have introduced techniques to solve this problem. A common solution is the use of Query Expansion with a thesaurus like the WordNet[4], or expanded with the use of an ontology (Xu and Croft, 1996; Schweighofer and Geist, 2007). Topic models e.g., Latent Dirichlet Allocation (LDA) and Latent Semantic Analysis (LSA) (Blei, Ng, and Jordan, 2003; Deerwester et al., 1990) as well as Distributional Semantic approaches (Sahlgren, 2008; Turney and Pantel, 2010) have been proposed. Distributional theory (Harris, 1954; Firth, 1957) hypothesize that words that live in close context have a similar meaning, therefore, it is possible that such techniques capture more semantics. Recently, Mikolov et al., (Mikolov et al., 2013b; Mikolov et al., 2013a) showed that distributional representation of words can be learned, such that words that occur in similar contexts lie really close in the vector space. The learned distributional representation is what is called *Word Embedding*, simply because each word is represented as a dense vector of real numbers which maps from a space with one dimension per word to a continuous vector space with much lower dimension. Mikolov (Mikolov et al., 2013b) further demonstrated the effectiveness of the word embedding with the *Word2Vec*[5] algorithm, which, when trained on a large dataset, is capable of inducing the semantic similarity and relatedness between the words such that two words that are close in meaning lie really close in the vector space, or put in another way, points to the same direction in the space. As an example, the words *'castle'* and *'mansion* would lie really close in the space and then presumed to be similar, thus, a problem like a synonymy is naturally overcome. An important question of concern is how to use the rich semantic knowledge from word embeddings to create a semantic representation for the query and document such that *Query-Document Mismatch* is overcome.

The third problem is that of *granularity* of retrieval. This is especially important in the case of *document management system*. First, IR should be about retrieving *facts* which are the precise response to any given query. The importance of document management system to the legal domain can never be overstated, and a system like the *EUNOMOS* (Boella et al., 2012a; Boella et al., 2016) which is a prominent legal document management system has raised the bar in this regard. IR systems like EUNOMOS retrieves any document that it considers to be related to the user query. EUNOMOS, in particular, uses WordNet (Miller, 1995) to expand terms in the query and then ranks the document based on how similar they are to the expanded query according to the Cosine similarity function (Boella et al., 2012b). This kind of similarity ranking, however, has bias for longer documents since they have a high probability of containing expanded query terms (Almquist, 2011; Hyman, 2012). More importantly, even though a document in its entirety is retrieved, a user may only be interested in a specific section or part of the document that is of specific interest. One of the characteristics of legal documents is that they are usually long, i.e. a document may contain tens of pages (Van Opijnen and Santos, 2017). The issue with most document management systems like EUNOMOS is that they take for granted the problem of *Information Overload* in the result that they produce for users. The peculiarity of legal documents, as regards their size/length make the issue of Information

[4] `https://wordnet.princeton.edu/`

[5] `https://github.com/RaRe-Technologies/gensim`

Overload to be important. For example, let us assume that the information need of a user is between the pages *5-7* of a legislative document which is *35* pages long. Let us also assume that an LIR system actually retrieves that particular document based on the user's query. Even though the retrieved document is relevant to the query, the user still has to manually read the retrieved document page by page in order to identify the section(s) of the document that is of interest. We call this process *manual information filtering*. The implication is that the retrieval result contains a lot of information or say, noise, which is irrelevant to the user (i.e., pages *1-4* and *8-35*), such that the user will still have to carry out manual filtering. We say that there is information overload when a user is given more information than is actually needed within a retrieved item, such that the user has to manually filter the information in order to identify the needed part. Given a query, a good LIR system would produce the relevant document(s) (e.g., the whole *35* pages), however, an effective LIR system would retrieve the relevant fact (e.g., pages *5-7*) which satisfy the user's information need. Such an ideal system will be able to automatically segment long documents into different sections and then match queries to section(s) of document instead of whole documents. This will amply solve the problems of information overload and document mismatch which happens because of the bias for longer documents.

An emerging IR trend is systems that provide high precision, simple and short but direct answer to a natural language question, some even in a conversational way. Examples include the personal digital assistants like Siri which was developed by Apple, or Cortana, which Microsoft bundled with their new operating systems not too long ago. It would be interesting to a lawyer who is gathering some information about a case to be able to query the system with a free text question like 'who presided over Zubulake v. UBS Warburg case' or 'where was the case of Zubulake v. UBS Warburg heard' and the system says 'Judge Shira Scheindlin' or ' United States District Court for the Southern District of New York.' respectively. These systems are generally called *Question Answering* systems. Question Answering (QA) system may rely on an external knowledge like the DBpedia (Auer et al., 2007), or rely on a self-contain sample of questions and their associated answers. The work of (Fader, Soderland, and Etzioni, 2011; Fader, Zettlemoyer, and Etzioni, 2014) are examples of the former which are classified as open domain, while the work of (Bordes et al., 2015; Bordes and Weston, 2016; Kumar et al., 2016) are examples of the latter, and are said to be the closed domain. As regards legal domain, QAKIS (Cabrio et al., 2012; Cabrio et al., 2013) also, operates over linked-data and has been deployed to answering questions on compliance in the Insurance industry. With the acclaimed success of Neural Networks-based QA systems (Bordes et al., 2015; Bordes and Weston, 2016; Kumar et al., 2016) which operates on synthetically generated data. A part of the work described in this thesis is to develop NN models for inducing semantic representation from legal documents, and to show how the induced representation may be applied for the QA task.

1.5 Thesis Question

Given the highlighted problems in Section 1.4, the main questions that the thesis aim to answer are:

RQ1- *How do we induce representation in order to capture the semantics of a user's information need?*

The second question this thesis will answer is:

RQ2- *How do we reduce the search space of information by ranking document based on their semantic relevance?*

The work described in the thesis aims at finding reasonable answers to these questions.

1.5.1 Research Approach

Most problems in the real-world can be solved in a holistic way, however, as regards real-world computing problems, no panacea provides a one-size-fits-all effective solution. As a matter of fact, it is often the case that when problems are thoroughly analyzed, we can easily identify different sub-problems which require specific but individual clear-cut solution. This approach to problem-solving is what is referred to as the *divide-and-conquer* in *Computer Science* parlance. The strong and appealing point of this approach is that a problem is divided into smaller parts which are then solved independently. Fortunately, it turns out that when individual solutions are combined, a robust and holistic solution to the bigger problem is obtained.

Our approach in this thesis is to adopt the divide and conquer solution paradigm. We highlight specific areas pertaining to our stated problems. Usually, we employ different kinds of legal dataset as demanded by the solution provided and the evaluation to be made. This thesis then provides an effective solution to each problem. When the solutions are viewed holistically, they address the three research problems highlighted in this thesis. Figure 1.3 is a pictorial representation of the tasks addressed in this thesis.

An attempt at providing fine-grained search solution is a system that divides a document into semantically coherent units called segments, these segments are then individually tagged with some concepts in order to allow a fine-grained conceptual search. In this approach, instead of retrieving the whole document, only specific part(s) of the document that is responsive to the query concept is retrieved (Adebayo, Di Caro, and Boella, 2017c). The Ensemble NN relevance model for the E-Discovery task is described in chapter 6. The work shows how important relevance signals are extracted from a document using the induced semantic representation from the Query. The other parts of this thesis also describes the NN models for inducing semantic representation and how this is applied to Question Answering. This part of our work is called Question Answering (QA) task (Bordes and Weston, 2016).

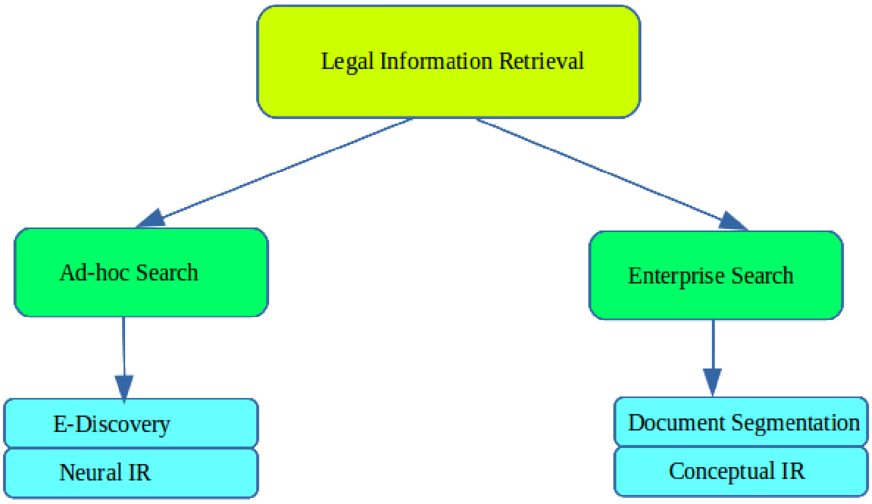


FIGURE 1.3: Overview of the tasks tackled in this thesis.

1.5.2 Research Goal

The overall research goal of this thesis is to induce semantic representation from different types of legal text at different levels of abstraction, which may then be applied to some IR tasks. Our goal is to explore and redesign the state-of-the-art NLP techniques and ML models for a variety of tasks in legal text understanding and processing. Our approach is to conduct a rigorous analysis of the documents which are the subject of this study, where applicable. Knowledge from this analysis is then used to develop systems that can induce needful semantic representation from the document. We addressed various search problems using different solution approaches. In some cases, we employed and combined existing NLP techniques and tools in a novel way while also developing new methods along the way. We are motivated by the recent exploits of *Deep Learning* (DL) approaches in various NLP tasks, and IR in particular, thus developing DL architectures that show effectiveness in the tasks that we address in the thesis. We measure the success of the different work described in the thesis either by evaluating using human annotated gold-standard, benchmarking our model against state-of-the-art models which have been evaluated on a different set of data (usually bigger), or by having human volunteers assess the result of our system.

1.6 Contribution to Knowledge

The significance of our work is how we induce semantic representation from different types of legal texts for the work described in this thesis. In particular, our approach shifts the IR task from matching by words to matching by meaning, using the induced semantic representation. We can situate the contribution to knowledge here according to the section of the thesis where the specific work is done. The significant contributions are presented below:

1. Concept-Based IR: We developed a concept to document mapping that works at the document segment level. The idea is to provide a fine-grained IR experience while solving two key problems here, i.e., user specificity, and granularity of retrieval. In the first instance, by allowing users to search for items using controlled concepts, users are freed of any worries about query formulation. Furthermore, since the approach operates at the level of document's semantics, i.e., the meaning of the concept, and the document part to be retrieved; the approach steps up from the keyword search to a semantic search. In the second instance, we proposed an algorithm that associates a concept not just to the document that it describes but to a specific part of the document. This not only produces a fine-grained result but also reduces the problem of information overload. The proposed method operates on two basic ideas; first is to use NLP approach to represent the meaning of a concept and the points of topic drifts in a text. Second is to associate the representation of a concept to a similar representation of a document. As a part of our work, we developed and utilized a topic-based text segmentation algorithm. Taking cognizance of the general nature of legal documents, the proposed algorithm divides a document into segments whose sentences share the same topics. The idea is based on the assumption that a document is a bag of topics, thus sentences with similar topics tend to cohere together. Using a list of *Eurovoc* concepts which are widely used for legal document indexing, the proposed system expands each concept using some NLP techniques in order to capture its meaning. The proposed system then maps a concept to a segment of the document that is most relevant to a query. To the best of our knowledge, we did not encounter in the literature, any system that offers this kind of fine-grained annotation for conceptual IR with respect to the legal text. This part of our work is partly adapted from (Adebayo, Di Caro, and Boella, 2016e; Adebayo, Di Caro, and Boella, 2016c; Adebayo, Di Caro, and Boella, 2017c)

2. Neural Network-based Relevance Model for E-Discovery: We propose a Neural Network-based relevance Model, a supervised classifier which determines whether a document is relevant to a query or not. Furthermore, the system learns to rank document according to their semantic relatedness to a given query using a weighted relevancy score that it learns to assign to document. NNs are already being employed for Adhoc IR, however, existing architectures either focus on query-document term matching at different scopes of the document, or the semantic similarity between the query the document texts. However, based on our observations, we discovered that E-Discovery is loosely dependent on the query terms and document terms relatedness. More importantly, the way the Request for Production (RFP) is normally presented gives no room for exact term matching, therefore, necessitating a new approach to representing both the query and the document. The proposed architecture is an Ensemble system, i.e., a combinatorial feature-rich Siamese architecture which uses component neural networks to extract multiple high-level semantic features from the query and document, and using another neural network to combine features obtained from the component neural networks. Furthermore,

the system also benefits from our newly introduced semantic query expansion approach which uses a fusion of a knowledge resource (WordNet) and semantic resource (Word Embeddings). The model typically overcomes language variability issues of polysemy and synonymy, especially since the focus is on semantic relatedness matching. The classification, ranking, and retrieval is performed end-to-end, and the model outperforms traditional bag-of-word based vector space model. The system has been evaluated on the 2010 and 2011 TREC legal track data.

3. Researchers usually initialize Neural Networks and encode words with pre-trained word vectors when applied to NLP tasks. Usually, this improves performance compared to when the network is initialized randomly, this is because pre-trained vectors readily capture syntactic and semantic information. It is usually expected that the size of data from which the vectors are obtained, coupled with the vector dimension, among other parameters usually influence how useful a pre-trained vector is. However, legal documents are strongly domain specific, and somewhat different to ordinary text, given that they do contain technical legislative terms. Similarly, it is our observation that the pre-trained word vectors are not created equally, and the data to be used to train a word embedding algorithm has to be domain-specific, provided it is to be used in a domain-specific task. In our work, we show our findings regarding this phenomenon, by showing that a superior performance can be obtained when the word vectors used are obtained from a custom data (e.g., legal documents) rather than a generic data (e.g., Wikipedia data) as revealed in our experiments. This is important especially for our work, where we show that our models capture legal nuances, hence, a good semantic representation of a document and query can be obtained.

1.7 Thesis Outline

In chapter 2, we discussed IR in depth. The desiderata of IR, the approaches to IR, as well as the general background knowledge needed for the later chapters. Chapter 5 of this thesis describes our concept-based semantic IR. We explain our notion of semantic annotation of document, the semantic similarity approach for matching segments of a document to the expanded concept etc. We also describe our topic-based text segmentation algorithm. In Chapter 6, we introduce our Ensemble Relevance matching Neural Network algorithm for the E-Discovery retrieval task. We describe the E-Discovery task and report our E-Discovery task evaluation using the TREC 2010 and 2011 legal track.

1.8 Publication

The work presented in this thesis has directly or indirectly benefited from the following published papers accepted and orally presented at peer-reviewed international conferences and workshops[6].

A. *Published Paper*:

1. (Adebayo, Di Caro, and Boella, 2017a): Siamese Network with Soft Attention for Semantic Text Understanding. In Proc. of Semantics 2017 Association for Computing Machinery (ACM)*.
2. (Adebayo et al., 2017): Legalbot: A Deep Learning-Based Conversational Agent in the Legal Domain. In LNCS (Springer) Proc. of International Conference on Applications of Natural Language to Information Systems (NLDB 2017)*.
3. (Rohan et al., 2017)[7]: Legal Information Retrieval Using Topic Clustering and Neural Networks. In Proc. of COLIEE 2017, collocated with ICAIL 2017 (Easychair)**.
4. (Adebayo et al., 2016a): Textual Inference with Tree-structured LSTM. In LNCS (Springer) Proc. of Benelux Artificial Intelligence Conference*.
5. (Adebayo, Di Caro, and Boella, 2016d): NORMAS at SemEval-2016 Task 1: SEMSIM: A Multi-Feature Approach to Semantic Text Similarity. In Proc. of ACL SemEval (ACL Anthology)**.
6. (Adebayo, Di Caro, and Boella, 2016a): A Supervised KeyPhrase Extraction System. In Proc. of Semantics 2016 Association for Computing Machinery (ACM)*.
7. (Adebayo, Di Caro, and Boella, 2016e): Text Segmentation with Topic Modeling and Entity Coherence. In Proc. of International Conference on Hybrid Intelligent Systems (HIS) (Springer)–*Awarded the Best Paper**.
8. (Adebayo, Di Caro, and Boella, 2016b): Neural Reasoning for Legal Text Understanding. In Proc. of Legal Knowledge and Information Systems - JURIX 2016: The 29 Annual Conference (IOS Press)*.
9. (Adebayo et al., 2016b): An approach to information retrieval and question answering in the legal domain. In Proc. of JURISIN 2016 Workshop**.

B. *Accepted and Awaiting Publication*:

1. (Adebayo, Di Caro, and Boella, 2017c): Semantic annotation of legal document with ontology concepts. Accepted for AICOL 2015 Springer LNCS Proceedings**.

[6] Conference papers are marked * while workshop papers are marked **.
[7] The first and second authors have equal participation

2. (Adebayo, Di Caro, and Boella, 2017b): Solving Bar Exams with Deep Neural Network. Accepted at 2ND Workshop on automated semantic analysis of information in Legal Text (ASAIL) 2017**.

1.9 Chapter Summary

This chapter lays a foundation for understanding the scope of our study. We highlighted some existing challenges which motivate our work. We discussed our research goal, focusing on our step-wise approach to information retrieval in the legal domain. The contribution to knowledge as well as a brief description of each work presented in each chapter. The datasets and the description, resources as well as our models and other tools that we used in this thesis are available upon request or at the moment through this link: https://www.dropbox.com/sh/vl8bhz0s20vbgy4/AABCd6O3uuwUQEYJMxJF9QJua?dl=0. In the future, it would be released via other public open source channels.

Chapter 2

INFORMATION RETRIEVAL AND RETRIEVAL MODELS

In Chapter 1 of this thesis, we stated that Information Retrieval (IR) seeks to determine the presence or absence of an information that is of interest to the user. For a system to give its users' the desired satisfaction, it must have a way of comprehending the exact need of its users and creating a representation of the need. An IR model tries to give a real-world representation of a user's need. This chapter will give a broad overview of what IR is, we will then discuss some of the important models of IR from the literature as well as the strategies for evaluating the performance of an IR model.

2.1 What is Information Retrieval?

One of the popular definitions of IR is the one given by Gerard Salton (Salton, 1968), who along with his students, was one of the pioneers of this challenging field of IR. Salton's definition is reproduced below:

> "Information Retrieval is a field concerned with the structure, analysis, organization, storage, searching and retrieval of information. "

Notwithstanding that the definition was given decades ago, it still presents the relevant idiosyncrasies of any modern IR system. Two things can be learned from this definition, the first being that for an item to be 'searchable', it has to be 'storable'. Secondly, the definition implies that the field of information retrieval is broad and not limited to a specific object type, perhaps, the reason why Salton refrained from explicitly specifying what an information is. In reality, information is that *need* which a user requires, be it music, text, videos and whatever object that can be organized or stored.

The work presented in this thesis focuses on textual document (or simply text). Right from the early days of the earliest retrieval system like the *SMART* system (Salton, 1971), as well as the pioneer work on *Vector Space* and *Boolean* model retrieval systems (Salton, Wong, and Yang, 1975; Salton, Fox, and Wu, 1983) till today, providing more efficient methodologies and techniques that scale with the increasing size of data for timely and improved retrieval of texts has been the focus of researchers (Salton and McGill, 1986; Croft, Metzler, and Strohman, 2010).

Retrieving information from a storage may be simplified if related pieces of information are arranged in individual records which have attributes (with related semantics) that link them to one another. A collection of such related record is called a *Database*, and users can retrieve information with the aid of a specialized language called *Structured Query Language* (SQL). The type of IR system that operates in this type of environment is referred to as the *Relational Database Management System* (RDBMS), and different types of data such as text, images, music etc. can also be retrieved from these systems. With RDBMS, a retrieval system uses syntax provided by the SQL to look for any specific information in a given record, e.g., the retrieval system may be asked to retrieve the content of the "Lastname" column in a "Student" table in a 'University' database. Because information is arranged in a methodical way in the order of their relationship, we say that this class of information is structured.

However, many text document collections are usually unstructured. First, the rate at which electronically stored information (ESI) are generated is unprecedented, in this scenario, it is difficult to format documents such that they can be arranged into tables and columns as with the RDBMS. Secondly, for any pieces of information to be arranged in a database, such information must have meta-data that could be used to group them. However, documents may not have such meta-data. Documents with such meta-data are said to be partly structured, e.g., they may have a structured header, however, the body is still unstructured and the header can only contain meta-data about the document and not exactly the information content of the document (Greengrass, 2000). Lastly, a free text does not have explicit attributes that provide the possibility of connecting the different parts of the text as required by an RDBMS.

The implication of all these is that there is no well-defined syntactic structure that the retrieval system might use to locate data with a given semantics. For instance, documents in a collection may have different topics. Even if we know that a document talks about the European Union (EU), we still do not know the specific thing about the EU that it is talking about, e.g., it could be EU open border in relation to the trade or open border with respect to immigration, which are two different things. If we agree that the documents at least talk about the EU or open-border, without regard for the specificity, there is still no explicit way of knowing where open-border or EU appears in the body of the text, i.e., the exact page, section, paragraph or sentence. This characteristic is what defines the 'unstructuredness', the absence of an explicit syntax for a document or a group of documents, or a lack of well-defined semantics which relate the syntactic structure of each document in case there is a partial existence of such a syntactic structure (Greengrass, 2000). A good IR system should be able to retrieve relevant information from an unstructured document collection, which is also the focus of this thesis.

As discussed in the previous chapter, most of the search activities can be categorized under *Desktop* search, *Enterprise* search, and *Web* search (Croft, Metzler, and Strohman, 2010). The Web search is the commonest, and millions of queries are issued by users to search the Internet using some search engines like Google, Yahoo, AltaVista etc. Because of the size of the Internet, building IR systems to perform at this scale in real-time requires some complex indexing techniques, especially since the objects to be retrieved are sparsely located on millions of servers all across the world. Also, the objects are typically unstructured. Some efforts are being put into the Semantic Web project (Berners-Lee, Hendler, and Lassila, 2001) originally conceived by Tim Berners Lee, which allows for embedding of more knowledge into web pages and documents in order to be more machine understandable and comprehensible. The introduction of personal digital assistants like the Microsoft's Cortana, or Apple's Siri has further simplified the Desktop search on the computer and Phone. Desktop search deals with files which are stored on a personal computer, examples include emails, music files or books etc. Enterprise search, on the other hand, is a kind of search done, for example, over a corporate Intranet. In this thesis, we categorize the E-Discovery task, which is Ad-hoc by nature, as a kind of Enterprise search.

2.2 The Goal of an Information Retrieval System

It is possible to analyze IR as a process or a chain of events that fully describes both the user and the system modeling part of the IR process. In this chain of events, we have the user activity and the IR system's activity (Baeza-Yates and Ribeiro-Neto, 1999). In particular, the user activity may be expressed in terms of two sub-processes, i.e., the *intention* and the *representation*. When a user needs an information, the first thing, i.e., the intention is to cogitate what his/her needs are, for the thought process is conceived in the mind. This cognitive duty includes a formulation of what the user likely expects as the right response to his need (relevant objects) and what he believes may not satisfy his need (irrelevant objects). Visualizing the picture of what is relevant or not helps in the representation sub-process, i.e., the user begins to formulate how to present his ideas of relevance according to his need in a way that the system can replicate his thoughts into reality. Obviously, the user may not in totality have a priori knowledge of all the information he is searching, but he has a modest knowledge of what he is not searching. This representation process is what is called *Query Formulation*. A query is, therefore, an express simplification of the user's thought about relevance as well as a specification of an information need. In the subsequent sections, we will describe various ways of representing queries.

Once a query has been formulated, an IR system must provide useful information which satisfies the query. An ideal IR system must have an understanding of the user query as well the documents in the collection in order for a meaningful match to be done. In Section 2.6, we provide a review of the most important models for query and document understanding as well as relevance/semantic matching. Apparently, relevance matching is not an easy task because documents in the collection can belong to different topics. Also, amidst other challenges, documents are usually expressed in unconstrained natural language (Greengrass, 2000). An ideal IR system must, therefore, possess some interesting characteristics. As we will see in later chapters, these desirable characteristics guide the design of the IR solutions proposed in this thesis.

2.3 Desiderata of an Information Retrieval System

The ultimate goal of an IR system is to produce relevant information which satisfies a user's information request. Determining whether a retrieved information is relevant to the query could be dicey, more importantly, since relevance itself is a subjective concept, e.g., is it topical relevance, semantic relevance etc. A good IR model must, therefore, have some explicit attributes which must guide its understanding of the kind of relevance it wants to model. Mitra and Craswell (Mitra and Craswell, 2017a) itemized some of these attributes. In this work, we use these attributes as the guiding template in the design of our proposed solutions.

1. **Semantic Understanding**: There are different sides to relevance, i.e. should it be about the exactness of terms that occurs in both the query and the document or more about other relative details or evidence which implies that a document says something in relation to another document or query? The latter is generally referred to as the '*aboutness*' of an information, e.g. query or document. Traditional IR approaches rely on the frequency of intersection of terms between a document and the associated query in order to judge relevance. While this count-based approach may not be the most ideal, it has performed reasonably well when enhanced with different weighing techniques like *IDF*, *TF-IDF*, and *BM25* (Salton and Buckley, 1988; Robertson and Zaragoza, 2009), and has been the fulcrum of approaches like the Boolean (Salton, Fox, and Wu, 1983; Lee, Chuang, and Seamons, 1997) and the Vector-Space models (Salton, Wong, and Yang, 1975; Lee, Chuang, and Seamons, 1997). The problem with this approach, however, is that it fosters the gap between how people conceive information and expresses same in natural languages. Human views the world in terms of the semantic and conceptual representation (Arazy, 2004), which is why we can easily understand that an *automobile* is conceptually similar to a *vehicle* or a *car*, etc. Counting word frequency would fail to realize this kind of similarity since words are naturally ambiguous, and two different words may express the same meaning. The result is that we relegate the essence of 'aboutness' when determining relevance. Also, 'order' and 'structure' of words are important in the way that human understands communication, which, unfortunately, are lost in the Bag-of-Words (BOW) based approaches. An ideal IR system must be able to distinguish between 'warm puppy' and 'hot dog', even

though the terms ’warm’ and ’hot’ are synonymous as is ’dog’ and ’puppy’ (Mitra and Craswell, 2017a).

2. **Robustness to rare inputs**: One of the guiding principles in Language Modeling (LM) is the Zipf’s law (Newman, 2005). This law states that, if t_1 is the most common term in the collection, t_2 is the next most common, and so on, then the collection frequency cf_i of the ith most common term is proportional to $1/i$:

$$cf_i \propto \frac{1}{i} \tag{2.1}$$

   What this means is that frequency decreases rapidly with the rank of a term, or put in another way, a few terms appear more prominent in a collection count while the majority terms are used sparingly. In other words, most of the words that a user might use in a query may be least known words or least used words in a document collection. An ideal IR model should be flexible and adaptive enough to rare words. A plausible way to do this is to consider performing an exact matching of the rare words in situations where a query word is not found in the exclusive vocabulary.

3. **Robustness to corpus variance**: An ideal IR system must not be too dependent or sensitive to the specificity of a corpus, otherwise, it may perform creditably when given documents from a related document to the one it was trained on while performing poorly when documents from another domain are involved. In real life, it is almost impossible to know a priori the kind of information or the kind of search that the prospective users might be conducting in the future. Machine Learning (ML) and especially Deep Learning (DL) based models may, for instance, be susceptible to such bias because they look for innate patterns in the data. This may cause such models to ’over-cram’ even the minutest details about the kind of data they are trained with while not being able to generalize enough across varieties of data. For instance, the authors in (Szegedy et al., 2013) show that by perturbing an input data, an equivalent Neural Network that was initially trained on the original data committed a lot of misclassification errors on the perturbed data. Also, Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) are showing incredible capacity whereby a neural network is generating fake data from an input data for another neural network which is deceived to believe that it is working with the original data. In the legal domain, a model should be effective, efficient and robust such that it is invariant or insensitive to the data it was trained with, and should function optimally irrespective of the kind of search being carried out, e.g., case law retrieval, E-Discovery ad-hoc search or any legislative document that may be the subject of search by a user.

4. **Robustness to variable document size**: Document comes in varying sizes, and it is common for an IR system to have some bias for longer documents at the expense of shorter ones. In fact, document normalization techniques like the TF-IDF (Salton and Buckley, 1988) and pivoted length normalization (Singhal, Buckley, and Mitra,

1996) were proposed to curtail this bias. Also, queries are usually short in comparison to documents (e.g., 1 to 10 words on the average), and even though techniques like Query Expansion (Xu and Croft, 1996; Buckley et al., 1995; Robertson, 1990; Voorhees, 1994) can help enrich query terms by including synonyms etc., it is still the case that longer documents have more terms in common to the query, and thus retrieved or ranked above shorter ones which may be more relevant. An ideal IR system must be robust to different data input size/length. In addition, it must be able to pinpoint the exact section of a document that is most relevant.

5. **Robustness to errors in input**: A good IR system must be robust to erroneous input. Users often make mistakes when entering their queries, thus changing the intent of their search. Likewise, a document could contain mistyped words, abbreviations, and other orthographic variations. An IR system must offer a way to *reformulate* a user's query into a way that it expresses the user's intent and can easily match the relevant documents. Word normalization techniques like *stemming* and *lower-casing* characters can help in this regard. Also pertinent to this is *spelling error correction* (Duan and Hsu, 2011) and *query re-writing* techniques (Guo et al., 2008; Brill and Moore, 2000; Carpineto and Romano, 2012).

6. **Sensitivity to context**: A claim of Compositional Semantics (Baroni, Bernardi, and Zamparelli, 2014; Grefenstette, 2013) is that the meaning of a sentence or phrase is composed of the meaning of its parts, i.e., the words. This is rightly so, to the extent that humans believe that words do not live in isolation. However, the meaning expressed by a word also depends on the meaning of its surrounding words or neighbours, otherwise known as the context. An example is the word *bank* which may refer to a *financial institution* when close to a word like *money* and *deposit*, or it can refer to a *hummock* when its context words include *river* or *water* etc. The implication is that an ideal IR must take cognizance of the context of each word in the query when computing the meaning of the query so as to exclude potential noise from the result set. Incorporation of word-sense disambiguation techniques (Yarowsky, 1995; Banerjee and Pedersen, 2002; Navigli, 2009) may help in this regard. This is particularly important in the legal domain where legislative terms are used, and where, as the saying goes, –'*the language of the law does not follow the law of language*'.

7. **Efficiency**: An ideal IR model must be able to scale-up with big data, no matter how humongous, and should offer its users a graded notion of relevance through ranking (Liu, 2009), especially in a recall-friendly domain like the legal field, where the system has to provide a lot of relevant documents (i.e., when the recall is favoured over precision). A fusion of filtering techniques that quickly eliminates grossly irrelevant documents from a list of candidate documents to be considered for review may also speed up the retrieval process. Query feedback techniques (Chen and Roussopoulos, 1994) can help in improving the effectiveness of the IR model.

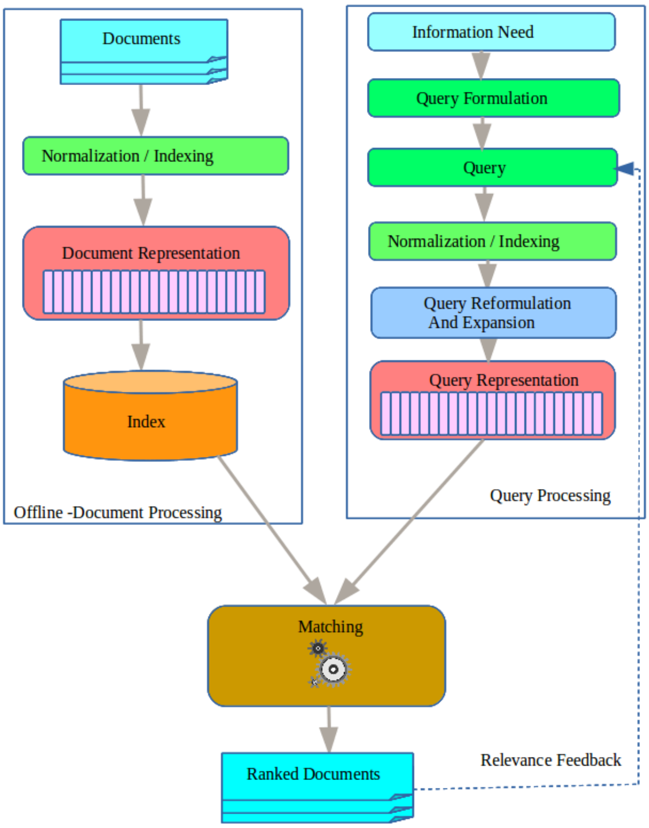


FIGURE 2.1: A General Information Retrieval Procedure.

2.4 Definition

IR systems are expected to provide relevant documents according to the information need of the user. The idea of what a *document* means in this regard can be ambiguous, i.e., is it a section of a document (as we will see in Chapter 5) or whole document as obtainable in the E-Discovery task that we discuss in Chapter 6. The general IR procedure is shown in Figure 2.1. It is thus important to specify what we refer to when a term is mentioned. Within the framework of this study, we define the two terminologies which concern the input to our retrieval systems.

2.4.1 Document

A document is a textual unit which is indexed as a candidate for IR system. The indexing which is a way of representing the document is mostly done off-line. A document is either relevant or not relevant. When the indexed candidate matches a user specification presented by a user through a query, the system returns this document. When we talk of retrieved items/documents in this thesis, three granularities are involved, depending on the task at hand. For instance, the end-result of our IR system in the E-Discovery task is a ranked set of whole documents. By whole we mean that each ranked document is a single entity or piece of evidence. In the conceptual annotation task described in Chapter 5, the retrieved item is indeed segments of document instead of whole documents.

2.4.2 Electronically Stored Information (ESI)

According to the Federal Rules of Civil Procedure (FRCP), ESI is information created, manipulated, communicated, or stored in a digital form requiring the use of computer hardware and/or software.

2.4.3 Collection

By collection, we refer to the corpus. For the ad-hoc task of E-Discovery, we employed the TREC legal track data. Every document in a collection is indexed for the IR task. Our solution matches the query with the documents in the collection and then yield a ranked list of the relevant documents according to their order of semantic relevance. Usually, the TREC evaluation requires that a top-k most relevant documents are produced, e.g., top 10,000 e.t.c. For other IR task that we present in this thesis, a separate collection is used as we shall describe in the subsequent chapters.

2.5 The Notion of Relevance

The concept of relevance has been been well studied by researchers (Park, 1993; Saracevic, 1996; Borlund, 2003). It is particularly important since we judge IR systems based on how successful they are in delivering relevant documents to the user. Borlund (Borlund, 2003) while referring to the work of (Schamber, Eisenberg, and Nilan, 1990) identified three views of relevance, i.e.,

- relevance is a *multidimensional* cognitive concept whose meaning is largely dependent on users' perceptions of information and their own information need situations
- relevance is a *dynamic* concept that depends on users' judgments of quality of the relationship between information and information need at a certain point in time
- relevance is a complex but systematic and measurable concept if approached conceptually and operationally from the user's perspective.

The term *multidimensional* reinforces the fact that relevance means different thing to different users, while by *dynamic*, Schamber et. al. (Schamber, Eisenberg, and Nilan, 1990) tries to express how perception might change with time. These views lay a ground for what Schamber (Schamber, Eisenberg, and Nilan, 1990) and subsequently, Borlund (Borlund, 2003) referred to as *situational relevance*, or put in another form, the *psychological relevance* (Harter, 1992). In this regard, relevance can be divided into two broad categories.

The first category leans toward the system-driven evaluation approach to IR, and it is called the *objective* or *system-based* relevance while the second one which leans toward the cognitive user-oriented IR evaluation criteria is called the *subjective* or *human/user-based* relevance (Borlund, 2003). In between these categories, Saracevic (Saracevic, 1975) identified five different manifestations of relevance. These manifestations are briefly described below:

- **System or algorithmic/logical relevance**: This relevance captures the relationship between the query and the retrieved document, i.e., do the query and the retrieved item express the same meaning? We liken this to the semantic relevance -a solution that is the ultimate aim of the thesis. This relevance must however not be situated to the concept of *utility*, which measures how useful the retrieved information is to the user; or *novelty*, which describes the proportion of relevant retrieved documents that a user just encountered (Lancaster and Gallup, 1973).
- **Topical**: This captures the *aboutness* of a retrieved document, and it is distinguished from semantic relevance. In this approach, a retrieved document that talks about *risk management for banks* may be appropriate for a user looking for documents regarding *regulatory compliance*. If a user is satisfied because the topic of the retrieved information relates to the topic of the information need, then such relevance is topical relevance.
- **Pertinence/Cognitive relevance**: This relevance emphasizes the notion of subjectivity, i.e., how a user perceives the information need and how impressive the retrieved item is to the user (Kemp, 1974). Moreover, it focuses on the amount of new information it is able to add to the existing knowledge of the user about his need.
- **Situational relevance**: This relevance captures the relevance of a retrieved item to a user based on the user's world-view. If a retrieved document changes the world-view of a user, then such document is situational (Wilson, 1973; Harter, 1992).
- **Motivational and affective**: This is a goal-oriented kind of relevance. If the retrieved information aids the achievement of a task or goal, then such kind of retrieved information is affective.

Our E-Discovery experiments follow the Cranfield IR evaluation model which has been used in many TREC tasks (Voorhees and Harman, 2005), and in particular, the legal track (Oard et al., 2008; Hedin et al., 2009; Cormack et al., 2010) where relevance is binary, i.e., given a query, a document can either be relevant or non-relevant. Another form of evaluation is through ranking, e.g., ordering relevance based on their likelihood probabilities of relevance. The Learning task of the TREC Legal track follows this style. Non-binary relevance is not easily evaluated with metrics like precision and recall. We follow strictly the algorithmic/logical relevance such that we explicitly model the semantic match between a document and a given query by incorporating some syntactic and semantic analysis of the query and the document in order to better capture their meaning and in particular, the user intent. This semantic matching also extends to other experiments other than the ad-hoc retrieval task for E-Discovery.

2.6 Information Retrieval Models

In the first chapter, we discussed that the earliest form of IR was through Keyword search, i.e., where prominent terms (most especially nouns) are used to index and then retrieve documents that explicitly contain the index terms. The *explicit appearance* condition imposed by this approach implies that the system oversimplifies the way a human understands and expresses language. Clearly, humans view and represent language in terms of concept such that a concept can express different meanings, i.e., words are usually ambiguous. This particular condition is what has been coined synonymy and polysemy, which are two recurring problems which influence the design of any IR system. Similarly, it is often the case that the meaning of a word may not be substantiated without considering the meaning of the neighbouring words. What this means is that keywords may not fully capture how we express our information need, and in the eventual case that it is used, irrelevant documents will overwhelm the relevant ones, if at all there is any(Baeza-Yates and Ribeiro-Neto, 1999).

One way of understanding how humans think of information and how we naturally express our thoughts in a language is through the use of models. Scientists often use models to explain a phenomenon, idea, or behaviour in the real-world (Hiemstra, 2009). It is often the case that such a behaviour cannot be experienced directly, thus, we can give a scientific model some hypothetical assumptions and in turn, the model can give a representation of such real-world experience. As discussed in the preceding section, an important theme to which every IR process revolves is the *relevance*. For instance, while someone's information need may prefer relevance in terms of topics, such that a document is relevant if it is 'about' something, that may not be sufficient for another person who sees relevance in terms of the semantic relationship or match, i.e., a relevant document must express the same 'meaning' as the query. A model may be employed to understand this variance in perception of relevance. The processes involved in IR also benefit from mathematical models which researchers have used over the years to codify how humans perceive relevance.

Baeza et. al. (Baeza-Yates and Ribeiro-Neto, 1999) gives a formal definition of IR model as a quadruple $\{\mathbf{D}, \mathbf{Q}, \boldsymbol{F}, \boldsymbol{R}(\mathrm{q}_i, \mathrm{d}_j)\}$ where **D** and **Q** are the representations of the document collection and the query respectively. The framework $\boldsymbol{F}$ captures the logical relationship between the document and the query representation, and finally $\boldsymbol{R}(\mathrm{q}_i, \mathrm{d}_j)$ is a ranking function which assigns a relevancy score to each document in the collection, based on its relationship to **Q** as modeled by $\boldsymbol{F}$. The success (or otherwise) of the model depends on $\boldsymbol{F}$, and if it fails, the ranking function may rank irrelevant documents higher than the relevant ones.

A lot of IR models have been proposed in the past and the improvements have been vertical, i.e., each succeeding model tries to overcome the weaknesses of the previous ones while of course retaining their strengths. Generally, these models can be classified into three categories, i.e., the Boolean/Set-theoretic model (BM), the Vector Space/Algebraic model (VSM), and the Probabilistic models (Baeza-Yates and Ribeiro-Neto, 1999). What differentiates these models from one another is what represents the framework ***F***. For instance, while this could be the vector of weighted terms of document and query and the linear algebra operations on the vectors for the VSM, it is the document representation and the manipulation with the set theory for the BM. The BM and VSM reiterates the general assumption of the bag of words (BOW) where the order or the syntactic connection between words is dismissed. Even though this may be too simplistic to model the semantics of natural languages, they are always a good first approximation and the fact that they have been effective over the years make them a good template which more powerful models can build on (Salton and Buckley, 1988; Lavrenko and Croft, 2001).

2.6.1 Boolean Model

The Boolean Model is a simplistic approach which relies on the set theory and Boolean algebra, i.e., the Boolean operators over strings that occur in a text. This model has been the approach of choice for many IR users especially in the legal domain because of its formalism, i.e., it allows queries to be specified by Boolean/Logical expressions, using operators *'AND'* known as the conjunction, *'OR'* known as the disjunction, and *'NOT'* which is the negation. The fact is that these expressions have precise semantics such that when combined users can flexibly express their information need by intervening the operators with the set of terms in the document collection. For example, the AND operator infers that a user wants all the document where the terms connected by the operator explicitly appears, e.g.,"Financial AND Regulation" produces documents where both terms appear. The OR operator, on the other hand, relaxes the condition as it produces the union of both terms. The NOT operator produces the documents that do not obey a logical expression or do not contain specific terms it was conjoined with, e.g., the query "Financial AND Regulation AND Compliance AND NOT Insurance" produces the document where the terms Financial, Regulation, and Compliance exist but where the term Insurance does not appear.

The retrieval framework of the Boolean model is represented below:

$$R(q,d) = \begin{cases} 1, & \text{if q is a term and present in document d} \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

Where the Boolean operators are modeled as shown in Equations (2.3) to (2.5) :

$$R^{OR}(q_1, ..., q_m, d) = \max_i R(q_i, d), \tag{2.3}$$

$$R^{AND}(q_1, ..., q_m, d) = \min_i R(q_i, d), \tag{2.4}$$

$$R^{NOT}(q, d) = 1 - R(q, d), \tag{2.5}$$

where i ranges from 1 to m, the number of arguments for each operator.

It is also possible to construct complex boolean queries by combining these basic operators and evaluating accordingly with the boolean algebra. Several refinements have been proposed to enhance this model. First, it is possible to target the query at a specific region or syntactic part of the document, e.g., title or abstract part may be targeted instead of the whole parts of the document. Second, the query may further be refined such that even in a particular region of the target, the search space is limited to a specific position, e.g., focusing on the first few words of the abstract rather than whole abstract. Third, we may use proximity operators (Mitchell, 1974) to further refine the search. For instance, with proximity operator, a user may specify how close in the document the operand terms must be to satisfy the query condition, such that the position offset between the terms is used as a condition for retrieval. The proximity operator applies both to terms as well as boolean operators (Greengrass, 2000). An example of this flexibility is to specify that some terms / sentences that satisfies a condition must be near or adjacent to another sentence that satisfies a different condition.

Croft et. al. (Croft, Metzler, and Strohman, 2010) opined that its main advantage is the predictable and easily explainable results. Also, the fact that document metadata may be substituted in lieu of word as operands to the logical operator makes it attractive. Also, it can quickly and effectively eliminate irrelevant documents from the search space. However, a drawback of this approach is that it does not allow ranked retrieval, i.e., it models the binary decision criterion whether a document is relevant or not relevant (Baeza-Yates and Ribeiro-Neto, 1999). It retrieves all the document that obeys the Boolean expression and in such a situation, it is difficult to pinpoint the best match for a query. Secondly, because it is index-based (i.e., terms are either present or absent and assigned corresponding weights w$_{ij}$ ∈ {0,1}), relevant documents are left out if they do not contain exact query terms. Therefore it can be considered as an exact match such that a word that is absent in the document receives zero weight. The work of (Salton, Fox, and Wu, 1983) introduced some normalizations to solve this specific problem with his extended Boolean model, important of which is the *p-norm* model. Here, operators make use of the weights (real number between 0 and 1) which are assigned to the terms in each document consequent upon the degree to which the given Boolean expression matches the given document, instead of the usual strict values 1(if term is present) or 0 (if term is absent).

The extended Boolean function from the *p-norm* is as below:

$$SIM_{AND}(d,(t_1,w_{q1})AND.....AND(t_N,w_{qN})) = 1-\left(\frac{\sum_{i=1}^{n}((1-w_{di})^p \cdot w_{qi}^p)}{\sum_{i=1}^{n} w_{qi}^p}\right)^{\frac{1}{p}}, (1 \leq p \leq \infty) \tag{2.6}$$

$$SIM_{OR}(d,(t_1,w_{q1})OR.....OR(t_N,w_{qN})) = \left(\frac{\sum_{i=1}^{n}(w_{di}^p \cdot w_{qi}^p)}{\sum_{i=1}^{n} w_{qi}^p}\right)^{\frac{1}{p}}, (1 \leq p \leq \infty) \tag{2.7}$$

Where p is a parameter for tuning the model and it takes on values between 1 and ∞.

Lastly, the fact that the operator allows for a flexible query does not take away from the fact that complex queries are often needed if very relevant documents are to be retrieved. The problem with this is that formulating such complex queries requires some expertise and experience for it over assume that users know exactly what they need. It is, however, often the case that users do not fully know how to express their need (Arazy, 2004). The use of search intermediaries who translate users need into a complex Boolean query may be required (Croft, Metzler, and Strohman, 2010). In a nutshell, with its logical structure, the burden is usually on the user to formulate an effective query, which novice or non-mathematical users find difficult to comprehend.

2.6.2 Vector Space model

It was Luhn (Luhn, 1957) in 1957 who opined that a simple way to retrieve relevant documents from a collection is to prepare a representation of the information need, in a way that it is similar to the documents wanted, and that if the representation of the documents in the collection is also made, a measure of similarity between the information need representation with those from the collection would yield a rank that may be used to identify the relevant ones. An implication of Luhn's approach is that each document and query needs to be indexed based on the collection of terms. For instance, if we represent a document by $\overrightarrow{d}$ = (d$_1$, d$_2$, ..., d$_m$) where each component d$_k$ (1 $\leq$ k $\leq$ m) is associated with an index term. If we also represent the query by $\overrightarrow{q}$ = (q$_1$, q$_2$, ..., q$_m$) such that the each query vector item q$_k$ references the same indexed word d$_k$ which carries a value between {1,0} depending on if the word appears in the document or query. Then, a vector inner product can tell us how similar both the document and the query are. The formula for calculating the inner product between the vectors of document and query is given in equation (2.8):

$$Sim(\vec{d},\vec{q}) = \sum_{k=1}^{m} d_k \cdot q_k \tag{2.8}$$

Both the document and query representation may be normalized further, such that, equation (2.8) is rewritten as shown in equation (2.9) below, which is equivalent to the cosine

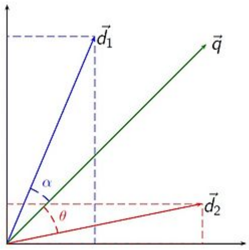


FIGURE 2.2: A Query and Document Representation in the Vector Space.

formula in equation (2.10):

$$Sim(\vec{d},\vec{q}) = \sum_{k=1}^{m} n(d_k) \cdot n(q_k), Wheren(v_k) = \frac{v_k}{\sqrt{\sum_{k=1}^{m}(v_k)^2}} \tag{2.9}$$

The vector space model builds on Luhn's approach by compensating for the inadequacies encountered in such a binary weighting approach (Salton, 1968; Salton, Wong, and Yang, 1975). The main improvement of Salton's Vector Space Model to the approach of Luhn is the use of real numbers (non-binary) for representing each term, and this is achieved by the introduction of a better term weighing scheme, e.g., the term frequency (TF), inverse document frequency (IDF), and the more robust one called the term frequency-inverse document frequency (TFIDF) (Salton, Wong, and Yang, 1975). The term weighting schemes enables us to compute the degree of importance of each term in the document in relation to every other terms such that we can represent that document as a vector of its term weights. In essence, we can compute the similarity between vectors representing a query and a document. Furthermore, both the document and the query can now be embedded in a high dimensional Euclidean space, such that, each term takes in a different dimension. Once we have the representative vectors of a document and the query, the next thing is to compute the similarity between these vectors. Instead of using the vector inner product, a more intuitive option is the cosine similarity method which measures the cosine of the angles between the norms of embedded query and document vectors, such that, the more orthogonal or farther apart two vectors are in the space, the lower the cosine of their angles, i.e., literally, higher cosine score between a query and a document means that they are more similar while a lower cosine value of the angles of two vectors means that the vectors are less similar. This is also the approach adopted for the SMART system (Salton, 1971) which in the past was a pioneer search engine. Figure (2.2) shows a visualization of a query vector and the vectors of two documents in Euclidean space. The cosine formula is given below in equation (2.10) :

$$Sim(\vec{d},\vec{q}) = \frac{\sum_{k=1}^{m} d_k \cdot q_k}{\sqrt{\sum_{k=1}^{m}(d_k)^2} \cdot \sqrt{\sum_{k=1}^{m}(q_k)^2}} \tag{2.10}$$

As earlier explained, the formula is given in equation (2.10) and it outputs a similarity score between 0 and 1. If the value is high then we say that the documents are similar. The fact that we have a graded score for each query-document pair makes it possible to actually produce a ranked result. For instance, if we sort the query-document similarity scores for all the documents in the collection in a reversed order, then, the most relevant pairs are placed on top of the queue. This is the idea of *ranked retrieval*, the fact that we can associate a relevancy value to each document in a way that we drastically reduce the problem of information overload.

The cosine similarity is a prominent choice for computing similarity, however, it does not come without some flaws. Salton (Salton and Buckley, 1988) for instance notes that cosine similarity has a bias against longer documents because it deals with multiple topics (Lee, 1995). Lee (Lee, 1995) suggested that a solution to the bias against long document is by calculating similarity using a hybrid of cosine-similarity result and the similarity score obtained when a term-frequency normalization technique is used. Other techniques for improving VSM is by breaking documents into sections/passages and calculating a separate similarity between the query and document passages. An aggregation of the similarity between passages of a document then becomes the similarity of the document with the query. Buckley and Salton (Buckley, Allan, and Salton, 1994) in particular introduced the concept of *global* and *local* similarity of a document to a query. The *global* similarity being the similarity of a whole document to the query while the *local* is the similarity of different parts of the document to the query. If two documents have similar global similarity score, then, the system switches to the local similarity such that the document that has a part/segment that is most similar to the query is selected.

The important decision to be made in this approach is what defines a section. To this effect, researchers have used different granularities in grouping document parts into section (Salton, Allan, and Buckley, 1993; Callan, 1994; Wilkinson, 1994; Kaszkiel and Zobel, 1997). A recent approach is to break a document into sections using topics as done in the TextTiling algorithm (Hearst, 1994; Hearst, 1997). As we will see in Chapter (5), our solution uses a more intuitive algorithm based on topic modeling to divide documents into sections (Adebayo, Di Caro, and Boella, 2016e). The implication of this is that we can properly explain why a section is relevant to a query than another section since each section contains coherent sentences (or paragraphs etc.) that talk about the same thing.

There are other techniques for computing similarity apart from the cosine formula. Korphage (Korfhage, 2008) introduced a similarity function shown in equation (2.11).

$$L_p(D_1, D_2) = (\sum_i |d_{1i} - d_{2i}|^p)^{\frac{1}{p}} \tag{2.11}$$

Where D_1 and D_2 are two document vectors, d_{1i} and d_{2i} are the components of D_1, D_2 respectively, and p is a parameter whose value ranges between 1 to ∞. The parameter determines the distance metric to be used between some available options, which include: *Euclidean distance*, *Maximal direction distance* etc. Other notable distance metrics are the *Dice* and *Jacquard coefficients* (Greengrass, 2000). The Dice's coefficient is computed by the formula given in equation (2.12).

$$Dice(D_1, D_2) = \frac{2w}{(n_1 + n_2)} \tag{2.12}$$

Here, w is the number of terms that is common to vectors D1 and D2. n_1 and n_2 are the numbers of non-zero terms in D$_1$ and D$_2$ respectively. The Jacquard's coefficient is computed by the formula given in equation (2.13).

$$Jacquard(D_1, D_2) = \frac{w}{(N - z)} \tag{2.13}$$

Where w retains the same property as in equation (2.12), N represents the number of distinct terms in the vector space, and z represents the number of distinct terms that are neither in D$_1$ nor in D$_2$.

Term Weighing Approaches

A document representation is usually obtained by splitting it up into individual terms which are then used to index the document and build up the vocabulary. Phrases or a conjoining of two or more contiguous terms, the so-called n-grams, are also a possibility. An intuitive way to capture the importance of each word in determining a document relevance is by associating each word with a weight which is a numeric value which shows its contribution to the meaning of the text. As a matter of fact, such weights are non-binary. The process of assigning this value to each term is called *term weighing*. There are various techniques for computing and normalizing term weights, and the reader is referred to (Greengrass, 2000; Manning, Raghavan, and Schutze, 2008) for a proper review. Specifically, the weight of a given term may be computed with respect to one of the following: 1) term frequency factor, 2) document frequency factor, and 3) document length normalization factor (Greengrass, 2000).

The simplest approach is to observe the number of time a term appears in a certain document. The idea of assigning weights to a term based on its frequency of occurrence is called *term frequency* weighing.

From observation, most documents follow the Zipfian law of distribution, such that some words appear more prominent while the geometric projection of other words that appear in that document is inverse. Conversely, a long document may contain some terms appearing once while a few appear hundreds of time. Past experiments have however shown that those repetitive few terms may carry less importance to the overall meaning of the document, e.g., the stop words and thus the raw count should be normalized. As shown in equation (2.14), the term frequency (tf) is calculated as a normalized count of the term occurrences in the document.

$$tf_{ik} = \frac{f_{ik}}{\max_i f_{ik}} \tag{2.14}$$

where tf$_{ik}$ is the term frequency weight of term k in the document D$_i$, f$_{ik}$ is the number of occurrences of term k in the document, and the maximum is computed over all terms in D$_i$. Salton and Buckley (Salton and Buckley, 1988) notes that in a collection, a term that appears equally in most documents in a collection may be less discriminating, and thus, may not be important to the meaning of any specific document. The *inverse document frequency* (idf) therefore put importance on words that appear prominently in a particular document but less frequently in others, and it is calculated by the formula in equation (2.15).

$$idf_k = \log \frac{N}{N_k} \tag{2.15}$$

where N is the total documents in the collection, n_k is the total number of documents where a term k occurs, and idf$_k$ is the inverse document frequency weight for the term k. Both *tf* and *idf* have their strength and weakness. An easy way of leveraging the weakness of one with the strength of the other is by combining them. This is called the term frequency-inverse document frequency of a document, and it is calculated as shown in equation (2.16).

$$tfidf_{ik} = tf_{ik} \times idf_k \tag{2.16}$$

Other term weighing approaches and their effectiveness can be found in (Robertson and Jones, 1976; Zobel and Moffat, 1998).

Latent Semantic Indexing

The traditional VSM described above though theoretically grounded has some limitations. First, the vectors are usually sparse and large since several terms will be missing in many documents, and this is because the dimension is defined by the indexed terms in a document collection. Also, it ignores the fact that users would like to retrieve based on concepts, and many words or document units that co-occur together may be grouped into topics. Lastly, it does not capture synonyms or polysemous relationship between words (Deerwester et al., 1990; Hofmann, 1999; Greengrass, 2000).

Deerwester (Deerwester et al., 1990) proposed a more plausible solution, i.e., the Latent Semantic Indexing (LSI), which captures the term-document relationship in a document collection. LSI is motivated by the distributional hypothesis that words that have similar meaning will always cohere in different texts (Harris, 1954; Turney and Pantel, 2010). Based on this, it uses a term-document matrix to capture the co-occurrence of words in the documents. The terms are then weighted using the *tf-idf*. Because the matrix is usually sparse, it finds a low-rank approximation by using the singular value decomposition (SVD) technique. The SVD decomposes the matrix into a low dimensional matrix and a column vector. It is then possible to compare both document and queries when they have been transformed into the low-dimensional space. The interesting thing here is that it captures a more semantic relationship that exists between words, e.g., words that have similar meaning now have similar co-occurrence features. Again as in the VSM approach, a separate vector is obtained for the document and the query, the similarity between these vectors can then be calculated using any distance metrics (cosine similarity especially) and the similarity score can be used to rank the most relevant documents to a given query.

2.6.3 Probabilistic Models

One of the earliest influence in the field of IR is the Library management system. Maron and Kuhns (Maron and Kuhns, 1960) while working on the algorithm for their 'mechanized' library system mooted the idea of *probabilistic indexing*, a technique that through statistical inference assigns a 'relevance number' to each document to show the probability that the document will satisfy the information need. Maron and Kuhns believed that when the relevance numbers for documents are reversely sorted, it will be easy to pick out the most relevant ones. Thus, the first probabilistic model for IR was birthed. Robertson (Robertson, 1977) extended their work and provided a more theoretically grounded solution which is not limited by the very 'mean' definition of relevance by Maron and Kuhns. Probabilistic models can be summarized by the argument of Cooper, which has been coined the *Probability Ranking Principle* (PRP) (Robertson, 1977):

> If a reference retrieval system's response to each request is a ranking of the document in the collections in the order of decreasing probability of usefulness to the user who submitted the request, where the probabilities are estimated as accurately as possible on the basis of whatever data has been made available to the system for this purpose, then the overall efficiency of the system to a user will be the best that is obtainable on the basis of that data.

We can therefore say that the main aim of these models is to estimate the probability that a document is relevant to a query. In fact, different techniques under this category of IR differ only in how they estimate these probabilities (Singhal, 2001).

The simplest of these models is the Binary Independence Model (BIM), or the so-called Okapi model (Robertson, 1977). The assumption here is that a document is associated to a random variable *R* which signifies relevance. *R* can take values 1 (relevant) or 0 (either relevant), such that each document is a binary vector over the vocabulary. We say that d ∈ {0,1}$^{|V|}$. As in the Boolean model, the term occurrence variables are conditionally independent, i.e., a term that appears in the document gets a value 1 in the document's vector and the term not found in a document gets a value 0. The model, therefore, seeks to identify if the probability for a document being relevant is greater than its probability of not being relevant, i.e., P(R=1 | d) > P(R=0 | d). Of course, in practice, we are particular about relevance than non-relevance, hence, P(R=1 | d) is used to rank the documents according to PRP. This is better modeled by Bayes theory as shown below:

$$P(R = 1|d) \overset{rank}{=} \frac{P(R = 1|d)}{P(R = 0|d)},$$

$$
\begin{aligned}
&= \frac{P(d|R=1)P(R=1)}{P(d|R=0)P(R=0)}, \\
&\overset{rank}{=} \frac{P(d|R=1)}{P(d|R=0)}, \\
&= \prod_{w \in V} \frac{P(d_w=1|R=1)^{\delta_w} P(d_w=0|R=1)^{1-\delta_w}}{P(d_w=1|R=0)^{\delta_w} P(d_w=0|R=0)^{1-\delta_w}}, \\
&\overset{rank}{=} \sum_{w:\delta_w=1} \log \frac{P(d_w=1|R=1)P(d_w=0|R=0)}{P(d_w=0|R=1)P(d_w=1|R=0)} \qquad (2.17)
\end{aligned}
$$

where d$_w$ is the occurrence variable, δ_w is 1 if a term w is found in the document and 0 if not found, and $\overset{rank}{=}$ denotes the rank equivalence. There are two possible scenarios where BIM may be used, i.e., when a relevance judgment is available and when it is not available. As we shall see in Chapter 6, relevance judgments are essential for the E-Discovery task or ad-hoc retrieval in general, and in our solution, we used the relevant judgments as the *learning examples* for the Neural Network algorithm. This learning examples contain both the positive and the negative sample of documents that are relevant to any given query. In essence, relevance judgments are human annotations, a Machine Learning algorithm observes some patterns from the example and uses the learned pattern to classify any given document as either relevant or not relevant. In a relevance judgment, there will be positive class (documents that are relevant given a query) and the negative class (non-relevant documents).

$$P(d_w=1|R=0) = \frac{nr_w + \alpha_n r}{TNR + \alpha_n r + \beta_n r} \qquad (2.18)$$

$$P(d_w=1|R=1) = \frac{r_w + \alpha_r}{TR + \alpha_r + \beta_r} \qquad (2.19)$$

where *TNR* and *TR* are the total numbers of non-relevant and relevant documents in the judgment. nr_w and r_w are the total amount of non-relevant and relevant documents that contain a term w, respectively. The smoothing parameters α and β prevents sparsity or zero probabilities and are often set at 0.5 and 0 respectively. Where there is no relevance judgment given, then, equation (2.20) is used to estimate the probability.

$$P(R=1|d) \overset{rank}{=} \sum_{w:\delta_w=1 \wedge w \in Q} \log \frac{N - df_w + 0.5}{df_w + 0.5} \qquad (2.20)$$

where df$_w$ is the frequency of document that contains the term w and N is the total number of documents in the collection.

An upgrade on the BIM is the 2-Poisson Model (Robertson, Rijsbergen, and Porter, 1980). The difference here is that a document is represented by a vector whose components are the frequencies of each term. Also, the dimension of the vector is the size of the vocabulary. The ranking function is calculated as shown in equation (2.21) :

$$P(R=1|d) \overset{rank}{=} \sum_{w:tf_w>0} \log \frac{P(d_w=tf_w|R=1)P(d_w=0|R=0)}{P(d_w=0|R=1)P(d_w=tf_w|R=0)} \tag{2.21}$$

where tf$_w$ is the frequency of a term w in a document. The term frequencies are assumed to be conditionally dependent.

The BM25 which stands for '*Best Match, version 25*' is an extension of the 2-Poisson Model and it was proposed by Robertson and Walker in 1994 (Robertson and Walker, 1994). The final version was first used at TREC-3 in 1995 (Robertson et al., 1995). Interestingly, the model is simple and performs creditably well with the right parameter settings (Robertson, Zaragoza, and Taylor, 2004). The ranking function is given below:

$$P(R=1|d) \approx \sum_{w \in Q \cap d} tf_{w,Q} \frac{(k_1+1)tf_{w,d}}{k_1((1-b)+b\frac{|d|}{|d|_{avg}})+tf_{w,d}} \log \frac{N-df_w+0.5}{df_w+0.5} \tag{2.22}$$

The Inference Network is another popular Probabilistic Model and it has been used in large-scale systems, e.g., the INQUERY (Callan, Croft, and Harding, 1992). Other variations of these models exist, and are well documented in the literature (Robertson and Zaragoza, 2009; Metzler, 2011).

2.6.4 Language Models

The goal of a Language Model (LM) is to estimate a probability distribution over lexical entities (most especially words) of a natural language, such that the statistical regularities of the language are obtained. Given a document collection, an LM assigns a probability of occurrence score to every word in the vocabulary (Croft, Metzler, and Strohman, 2010; Croft and Lafferty, 2013). In information retrieval, the goal is to estimate the probability of generating a query from the document model (Ponte and Croft, 1998), put in another way, LM seeks to establish the likelihood of a query and a document being generated by the same language model, provided that the model that generates the document is known, without recourse to whether the model that generates the query is known or not known (Liu and Croft, 2005). It has been well applied to IR (Ponte and Croft, 1998; Hiemstra, 1998; Song and Croft, 1999). The common framework is to use n-grams, i.e., unigram, bigram and trigram models. The unigram model assumes term independence. Conversely, the probability of generating a document or query is obtained as the product of the probabilities of all the constituent terms. Assuming a sentence S is a sequence of k words such that:

$$\mathrm{S} = \mathrm{w}_1, \mathrm{w}_2, \mathrm{w}_3, \ldots, \mathrm{w}_k$$

then the model that generates S is given below:

$$P_n(S) = \prod_{i=1}^{k} P(w_i|w_{i-1}, w_{i-2}, w_{i-3}, ..., w_{i-n+1}) \tag{2.23}$$

Here, we assume that n = 1. When n = 2 or n = 3, then, it is a Bigram and a Trigram model respectively. Unlike the Unigram model, the Bigram and Trigram models capture the contextual information such that the probability of a word is dependent not only on its probability but also on the probability of prior words. Surprisingly, Unigram model, which is a simplification of the BOW works well for IR.

The *Query Likelihood* (Ponte and Croft, 1998) was the earliest approach to applying LM to IR task. Given a query Q, using a Bayesian estimate, this approach rank documents according to the likelihood that the query is a representation of the text. The computation is done as shown below:

$$\begin{aligned} P(Q|D) &= \prod_{q \in Q} P(q|D), \\ &= \prod_{q \in Q} \int_{\theta_D} P(q|\theta_D)P(\theta_D|D), \\ &= \alpha \prod_{q \in Q} \int_{\theta_D} P(q|\theta_D)P(D|\theta_D)P(\theta_D) \end{aligned} \tag{2.24}$$

The θ_D is a multinomial distribution over the vocabulary, and we say that it is the model that generates the document. We introduce a Bayesian smoothing P(θ_D), which is, of course, a Dirichlet (Zhai and Lafferty, 2001). The probability estimate of a word given a collection and the probability that a word is generated from a document is shown in equations (2.25) and (2.26) respectively:

$$P(w|C) = \frac{cf_w}{|C|} \tag{2.25}$$

$$P(w|D) = \frac{tf_{w,D} + \mu P(w|C)}{|D| + \mu} \tag{2.26}$$

here, C is the documents collection, and ∣C∣ is the vocabulary size. $tf_{w,D}$ and cf$_w$ is the frequency of a word in document D and collection C respectively. Given the Dirichlet parameters αw = μP(w∣C), μ is a hyperparameter for the model, and its value is usually set to be 2000. Documents can be ranked accordingly following the equations below:

$$\begin{aligned} P(Q|D) &\overset{rank}{=} \sum_{q \in Q} \log \frac{tf_{w,D} + \mu P(w|C)}{|D| + \mu}, \\ &\overset{rank}{=} \sum_{q \in Q \cap D} \log \left(1 + \frac{tf_{w,D}}{\mu} . \frac{|C|}{cf_w}\right) - |Q| \log(|D| + \mu) \end{aligned} \tag{2.27}$$

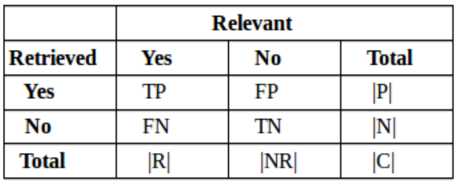
The unigram model embodies the standard *tfidf* formula but incorporates a more robust document normalization. Moreover, it also suffers from the weaknesses of the *tfidf* and BM25 techniques.

2.7 Evaluation of Information Retrieval Systems

It does not serve any good if we have a model without ascertaining how good it performed, and if it compares favourably with other retrieval techniques. Most times, the complexity of a model is not commensurate with its performance, and there have been cases where simple baseline like the BOW outperforms a more theoretically complex model. In order to ward off any subjective assumption about a model, we need an objective way of gauging the performance of IR models. Available metrics can be grouped under two different paradigms, i.e., effectiveness measure and the efficiency measure (Croft, Metzler, and Strohman, 2010). Furthermore, the specific IR metric and how it is used depends on the kind of retrieval activity that is being carried out, e.g., either unranked or ranked retrieval. For the ranked retrieval solution presented in this thesis, we followed the Cranfield evaluation standard (Voorhees, 2001; Voorhees and Harman, 2005) which has been adopted for TREC[1] retrieval tasks. The datasets for TREC tasks have similar properties to other popular ones like *GOV2*, *CACM*, *CLEF*, *NTCIR* and *AP* collections.

In order to measure the effectiveness of an IR system, there must be a *test collection* which contains some queries with their associated relevant documents. In an ad-hoc retrieval task like the one we present in chapter 6, the test collection must contain documents, some information needs (probably expressed as queries or topics as regards *TREC*), and the relevance judgment. The relevance judgment, which is also called the *gold standard* or the *ground truth* is a binary assessment of a query-document pair, which signals whether the document is relevant to the query. In a supervised machine learning approach, part of the relevance judgment is usually used as the *seed set* or training sample to feed a classifier with. This is usually called *predictive coding* in E-Discovery (Cormack and Grossman, 2014). It is important that the test collection is of considerable size so as to cater for any randomness in the result. In the scenario where there is no explicit relevance judgment, it is possible to use human assessor to directly evaluate the relevance of the retrieved document given a query. In order to ensure the integrity of the evaluation, it is important that the IR system must not have any privy knowledge of any sample from the test collection. In machine learning approaches, we usually set apart a portion of example document for optimizing the parameters of the system. This portion is often called the *development set*, and only it and the train set may have been seen by the system before the evaluation is carried out. It is possible to also differentiate evaluation based on whether the retrieval is ranked or not. The *TREC* Legal track dataset used in the solution described in the Chapter 6 requires a ranked answer. Generally, this kind of task is recall-oriented, which

[1] The reader is referred to `http://trec.nist.gov/overview.html` for an overview of *TREC* tasks.



	Relevant		
Retrieved	**Yes**	**No**	**Total**
Yes	TP	FP	|P|
No	FN	TN	|N|
Total	|R|	|NR|	|C|

FIGURE 2.3: A Contingency Table for Relevance.

means that the cost of missing out a relevant document is higher than when an irrelevant document is produced. As we will see in the Chapter 5, in an unranked retrieval, users are mostly interested in a system which retrieves a precise or an exact document(s) that satisfies the information need.

Assuming that a document collection C contains the set of relevant (denoted R) and non-relevant (denoted NR) documents such that the task of the IR system is reduced to a simple 2-class binary classification N or NR. We can also say that the system assumes that the retrieved documents belong to the positive class (denoted P) while those that were not retrieved belongs to the negative class (denoted N). This understanding is better visually represented as a confusion matrix as shown in table 2.3, where we can view the matrix as separating the collection C into four partial sets, i.e. the True Positive TP which is the number of relevant documents in the C that the system correctly classified as relevant, True Negative TN which is the number of irrelevant documents in C that the system correctly classified as being irrelevant, False Positive FP which is the number of irrelevant documents in C that the system incorrectly classified as relevant, and lastly, the False Negative FN which is the number of relevant documents in C incorrectly classified as irrelevant. An ideal system would ensure that items in its positive class are actually those labeled to be relevant and vice versa. Evaluation metrics usually measure efficiency in terms of the misjudgment of the system as regards these four partial sets. Below, we discuss the metrics commonly used in both ranked and unranked retrieval evaluation.

2.7.1 Precision

When an IR system retrieves some documents in response to the query, it is possible that not all the documents retrieved are relevant. The fraction of the retrieved documents that are relevant is referred to as the Precision (Baeza-Yates and Ribeiro-Neto, 1999).

$$Precision = \frac{\text{Number of relevant documents retrieved}}{\text{Number of documents retrieved}},$$

$$= P(relevant|retrieved),$$

$$P == \frac{TP}{TP + FP} \tag{2.28}$$

Two variants of the precision metric used in ranked retrieval are *Precision at k* and the *R-precision*. Unlike ordinary precision which accounts for exactness at all levels of recall, *Precision at k* limits the precision to a specified low recall level, i.e., say 20 or 50 documents.

Where *k* is the specified value, e.g., 'precision at 50'. An interesting feature of this metric is that it cares less about the size of the relevant documents in the collection, however, it may not give an approximate evaluation, the reason being that the total number of relevant documents for a query impacts on the precision at *k*. The *R Precision* gives a better approximation for it adjusts for the size of the set of relevant documents. Overall, it relies on the knowledge of relevant documents (*Rel*) from which the precision of the top *Rel* documents returned by the system is calculated.

2.7.2 Recall

Precision measures the exactness of a system and may not be the best metric since it does not consider the actual documents that are relevant. Recall on the other hand measures the completeness since it considers the relevant documents retrieved in proportion to the total documents that are actually relevant in *C*.

$$\begin{aligned} Recall &= \frac{\text{Number of relevant documents retrieved}}{\text{Number of relevant documents in the collection}}, \\ &= P(retrieved|relevant), \\ R &== \frac{TP}{TP+FN} \end{aligned} \tag{2.29}$$

2.7.3 F-Measure

The F-Measure combines the benefit of the Precision and the Recall into one. This is good because while some IR tasks favour precision, others would be better evaluated using recall. As an example, it would be delusional to assume that a system is optimal if it achieves 100% recall simply because it retrieves all the documents in a collection while obtaining a very poor precision score, or if the system achieves 100% precision score simply by retrieving just one document (which fortunately is relevant) out of a possible 50 documents, and consequently achieving 2% recall. Moreover, while recall grows with the number of documents retrieved, we expect that a good system achieves an increase in its precision inversely to the growth in the number of documents being retrieved. The F-measure, therefore, strikes a balance by forcing the two to trade off their rigidity against one another. It is computed as the weighted harmonic mean of the Precision and the Recall (Croft, Metzler, and Strohman, 2010).

$$F = \frac{1}{\alpha\frac{1}{p} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R}, \quad where \quad \beta^2 = \frac{1-\alpha}{\alpha} \tag{2.30}$$

where α takes a value between 0 and 1, while β^2 takes a value between 0 and ∞. β is a weighting parameter for the precision and recall. When β > 1, recall is favoured over precision and vice versa with a lower value for β. The *balance F measure*, so-called F_1

because the value of β = 1, is derived from the equation below:

$$F_{\beta=1} = \frac{2PR}{P+R} \tag{2.31}$$

2.7.4 Mean Average Precision

The *Average Precision* (AP) calculates the mean of the precision obtained for the top-k ranked documents existing after each relevant documents. Assuming the relevant documents for a query q$_i$ ∈ Q is {d$_1$, d$_2$,...., d$_{m_j}$} and R$_{jk}$ is the set of ranked retrieval results from the top result until the document d$_k$, then, AP is calculated based on the formula below:

$$AP = \frac{1}{m_j}\sum_{k=1}^{m_j} Precision(R_{jk}) \tag{2.32}$$

The Mean Average Precision (MAP) is the mean of the score obtained in equation (2.32), when averaged over the set of queries (Manning, Raghavan, and Schutze, 2008).

$$MAP = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{1}{m_j}\sum_{k=1}^{m_j} Precision(R_{jk}) \tag{2.33}$$

Because MAP weighs each query equally, it is most preferred for ranked retrieval domains like web search, etc.

2.7.5 Normalized Discounted Cumulative Gain

Normalized Discounted Cumulative Gain (NDCG) is mostly applicable where relevance is not restricted to the binary case of relevance or non-relevance. It is mostly used to evaluate machine learning based IR systems. It is similar to the *precision-at-k* in that evaluation is also done over a specified *k* of top search results. For a set of information need *Q*, if *R(j,d)* is the relevance score assigned by human assessor for a document *d*, given a query *j*, the NDCG score is calculated as below:

$$NDCG(Q,k) = \frac{1}{|Q|}\sum_{j=1}^{|Q|} Z_{kj}\sum_{m=1}^{k}\frac{2^{R(j,m)}-1}{\log_2(1+m)} \tag{2.34}$$

where Z$_{kj}$ is a normalization factor that conditioned the NDCG score at *k* for a query *j* to be 1.

2.7.6 Accuracy

The accuracy is calculated with the formula below:

$$Accuracy = \frac{TP + TN}{(TP + FP + FN + TN)} \tag{2.35}$$

2.8 Approaches for Improving Effectiveness

Many times, a theoretically grounded approach for IR may not live up to its potential in terms of performance. Several reasons could be adduced to such a situation. For example, the BOW relies on words as lexical units. If the same word appears in a document in more than one orthographic form, then each of the words is indexed as a separate term. Ideally, a system should reduce words like *went*, *go*, *gone*, etc to a single form. This kind of normalization is usually referred to as *stemming*. Also, where applicable, *parts-of-speech* (POS) tagging may also be done, for example, to identify POS like nouns and verbs which may carry more informative weights in a document. Even when these techniques are fully integrated, performance may still not be optimal owing to the fact that the query is usually a collection of a small piece of terms in comparison to the documents which are usually hundreds of order of magnitude higher. Below, we discuss some techniques usually used to improve the performance of IR systems. Some operate by enriching the query with more terms. The belief is that such an enrichment would incorporate more important words for the query to be able to match the relevant documents. However, as we explain below, each of them has its strength and weakness.

2.8.1 Relevance Feedback

Relevance Feedback (RF) is a technique that uses user-derived knowledge about the relevance of a document to improve the retrieval process (Salton and Buckley, 1997; Manning, Raghavan, and Schutze, 2008). The knowledge used to improve retrieval could be derived implicitly or explicitly. The technique is an iterative process whereby an IR system accepts a query, produce some documents which it believes to be relevant to the user, the user checks the produced result and accepts those that are relevant and reject those that are not. The IR system then uses this new knowledge in order to derive a better representation of the query and consequently, a better result. The Rocchio algorithm (Rocchio, 1971) which was introduced in the SMART system is a prominent technique. As shown in equation (2.36), the goal is to obtain a query vector that maximizes the similarity with relevant documents while minimizing the similarity with irrelevant documents. In the equation, D$_r$ and D$_n r$ represents the set of relevant and non-relevant documents and the q$_0$ in equation (2.37) represents the original query vector. The *sim* function could be any Euclidean Distance, for instance, the cosine similarity function in equation (2.10).

$$\vec{q_opt} = \underset{\vec{q}}{argmax}[sim(\vec{q}, D_r) - sim(\vec{q}, D_nr)],$$

$$\vec{q_opt} = \frac{1}{|D_r|} \sum_{\vec{d_j} \in D_r} \vec{d_j} - \frac{1}{|D_{nr}|} \sum_{\vec{d_j} \in D_{nr}} \vec{d_j} \tag{2.36}$$

Rocchio included three weight parameters γ , β and α which are assigned to each term as shown in equation (2.37).

$$\vec{q_m} = \alpha \vec{q_0} + \beta \frac{1}{|D_r|} \sum_{\vec{d_j} \in D_r} \vec{d_j} - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d_j} \in D_{nr}} \vec{d_j} \tag{2.37}$$

Other techniques that have been used for RF are the probabilistic models like the Naive Bayes, based on the probabilistic IR models (Robertson and Zaragoza, 2009), and the Neural Network based relevance feedback (Crestani, 1994). Salton (Salton and Buckley, 1997) notes that probabilistic RF does not perform well as their conventional counterparts. It is pertinent to also mention the *pseudo-relevance feedback*, usually called the *blind feedback* which assumes that the *top-k* retrieved documents are relevant, and terms from these documents are re-inserted to boost the original query terms. The problem with this approach is that a lot of noise could be inserted into the query which will lead to the system retrieving a lot of irrelevant documents. In general, RF techniques favour recall over precision. Also, they do not solve vocabulary mismatch problem along with word inflection issues (Baeza-Yates and Ribeiro-Neto, 1999).

2.8.2 Query Expansion

Natural languages are ambiguous and we can express a single concept in different ways. This unconstrained way of using words by humans implies that IR systems have to grapple with synonyms and polysemous words if a true understanding of the query and the document is to be achieved. Most especially, synonyms along with word inflections, e.g., plural forms like 'boys' compared to 'boy' often decreases recall. Likewise, polysemous words often leads to drastic reduction of recall.

Query Expansion (QE) is a query boosting technique where words or phrases that are semantically similar to the original query terms are used to expand the query, expand the scope of a search, and resolve term mismatch problems (Carpineto and Romano, 2012). This process can be fully automatic or semi-automatic in which case, human interaction in suggesting probable words to be included is needed (Croft, Metzler, and Strohman, 2010). The conventional approach has been the use of ontology and thesaurus (e.g., MeSH, Eurovoc, WordNet) to identify new words to be included in the query. For example, WordNet is an English thesaurus that contains synonyms (synsets) for each word, the most similar synsets to a query word might be included. A review of ontology-based automatic query expansion is provided in (Bhogal, MacFarlane, and Smith, 2007). Some researchers have also used knowledge from external corpus like the Wikipedia to expand queries (Li et al., 2007; Arguello et al., 2008). In any case, the co-occurrence of terms must be well analyzed and the words that are most appropriate considering the context or topic of the query must be selected (Croft, Metzler, and Strohman, 2010).

As we will see in chapter (5), instead of relying on the WordNet or an external corpus like the Wikipedia, we draw knowledge about semantic similarity from word embeddings which are trained on billions of words. The use of word embedding is intuitive since it naturally incorporates the contextual information in summarizing the meaning of a word. We complement this with the use of a thesaurus, i.e., Eurovoc, which is often used in the legal domain. The combination of our approaches enables us to be able to expand a concept and associate it not only with whole documents in the collection but to a specific portion of the document that the expanded concept is most semantically related.

2.8.3 Query Reformulation

Query Reformulation (QR) is the process of altering, refining, re-writing or transforming a query to another form without losing the original meaning, such that the new query can match relevant documents. Solutions include spelling correction, stemming, query segmentation and reduction (Li and Xu, 2014). The challenge in QR is to avoid *topic drift* so that the transformed query can match relevant documents. For example, rewriting the query- 'arms reduction' to 'arm reduction' could be misleading and totally perverse the meaning. Spelling correction (Brill and Moore, 2000) is particularly important for web-based queries and it is not so relevant to the type of retrieval performed in the E-Discovery task since experts carefully formulate topic/query. Query segmentation (Li and Xu, 2014) on the other hand may be useful in this regard because building phrasal units from the topic/query terms may lead to an improved recall.

2.8.4 Word-Sense Disambiguation

Even though techniques like QE and RF can partially help to resolve ambiguities in natural languages, they are most suited to obtaining synonyms of words. Polysemy is when a word has more than one meaning, and it is a frequent occurrence in most natural languages. Humans can easily understand the meaning of a word based on its context. Word-sense disambiguation (WSD) techniques aim at properly assigning the appropriate meaning to a word in a text (Bakx, Villodre, and Claramunt, 2006) and has been shown to improve IR systems performance (Uzuner, Katz, and Yuret, 1999).

Particularly as regards IR, WSD may be used to address the topic drift problem. For instance, it can be used in combination with QE and RF to improve an IR system performance. As an example, if the word bank appears in a query word with the intended meaning as 'the bank of a river', a QE system may look for synonyms of the word from a thesaurus, e.g., the WordNet. Since several senses of the word 'bank' can be found in the thesaurus, it may be difficult to know which one to select. Also, selecting all the synonyms, e.g., those related to 'bank' as a noun (e.g., a financial institution) or as a verb (e.g., to 'bank on' something which means to 'rely on') would have introduced unnecessary noise to the query terms. In this scenario, a WSD may be used initially to understand the sense of the word, that is, identify that the 'bank' referred to in the example refers to the river, and then only the synonyms for that specific sense are retrieved for the query expansion.

2.9 Word Embedding

Several natural language processing tasks in the past use vector space to encode words. The weaknesses of this approach, such as sparsity, high dimension and being unable to capture distributional features have been well researched in the literature (Turney and Pantel, 2010; Mikolov et al., 2013b) as we have discussed in the preceding sections. An Embedding is a representation of some items in a specified dimensional space such that the attributes, properties, and relationships between the items are better captured. A word embedding $\boldsymbol{W}$: words $\rightarrow \mathbb{R}^n$ is a parameterized function mapping words in some language to high-dimensional vectors. Most importantly, these embeddings have low user specified dimensions, usually between 50,100, 200, 300 and 500.

Embeddings may be induced through a lot of techniques e.g., the term-feature matrix factorization based on Latent Semantic Analysis (LSA) (Deerwester et al., 1990). Another approach is to use Neural Networks which learn to predict the contextual features of a given term (Bengio et al., 2003). For instance, **W** learns from a randomly initialized vectors for each word and optimizes its errors such that it is able to generate meaningful vectors for each word. Mikolov and his colleagues further demonstrates the practicability of this approach with the introduction of the Word2Vec algorithm (Mikolov et al., 2013b; Mikolov et al., 2013a), where they trained two variants of their algorithm (i.e, the skip-gram and the continuous bag-of-word (CBOW)) on a big dataset, and used the neural network to be able to predict the missing words in a sentence. They showed that the embedding incorporates very rich syntactic and semantic information about the words to the extent that an algebra computation may be performed on these representations, and a meaningful result would be obtained, e.g. **vector**('King') - **vector**('Man') + **vector**('Woman') = **vector**('Queen') (Mikolov, Yih, and Zweig, 2013). Figure 2.4 shows how *king* points to the direction of *queen* and *man* to *woman*. More importantly, vectors of individual words in a sentence can be combined to obtain the meaning of the sentence. Quoc and Mikolov demonstrated this with the paragraph vectors (Le and Mikolov, 2014). Researchers like Baroni (Baroni, 2013) and Grefenstette (Grefenstette et al., 2014) have also done extensive work on compositional semantics where various composition operators have been studied with regards to their performance. Because of these interesting properties, a lot of natural language processing research has since incorporated these neural word embeddings. Furthermore, new techniques to generate word embedding have been

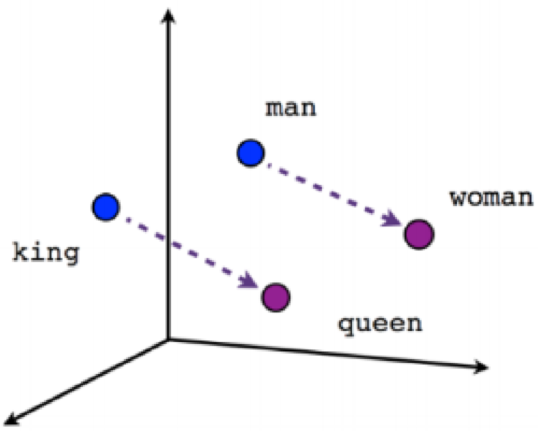


FIGURE 2.4: A 2-D embedding visualization showing how the related terms lie close in the vector space

proposed (e.g., see Joulin et al., 2016). As regards IR, the significance of word embedding has been well studied (Mitra et al., 2016; Mitra and Craswell, 2017b). As we will show in the following chapters, we have employed word embedding in many of the solutions described in this thesis. In particular, we have utilized the GloVe word embedding (Pennington, Socher, and Manning, 2014) and where necessary, we have used the Word2Vec algorithm (Mikolov et al., 2013b) to induce embedding from some collection of data.

GloVe is an acronym for *Global vectors for word representation*, and it is an unsupervised learning algorithm for obtaining vector representation for words. The algorithm was trained on an aggregated global word-word co-occurrence statistics from a corpus. To be specific, for most of our experiments, we utilized the one trained on 840 billion words (Common Crawl) with the embedding matrix dimension = 300. In other instances, we have trained the Word2Vec algorithm on a corpus of Legal texts (Adebayo et al., 2016b) and used it our experiments. We assume that the embedding obtained when an algorithm like the Word2Vec algorithm is entirely trained on a set of Legal documents may properly capture the nuances and the semantics of Legislative terms. This assumption has been validated in our previous work (Adebayo et al., 2016b). The obtained representations showcase interesting linear sub-structures of the word vector space, and it is more useful for any semantic task because of the size of the data it has been trained on.

2.10 Machine Learning and Information Retrieval

The main idea of Machine Learning (ML) is to develop algorithms that learn autonomously, and improve with experience without being explicitly programmed (Bishop, 2006). ML may be classified according to the underlying learning strategies, the representation of knowledge or skill acquired by the learner, and the application domain where the system is being used, e.g., whether it is a classification, clustering or ranking task. The authors in

(Michalski, Carbonell, and Mitchell, 2013) articulated different types of learning strategies, such as rote learning or direct implanting of new knowledge, learning from instruction, learning by analogy, and lastly, learning from examples. The latter is sometimes referred to as *supervised learning*.

Supervised learning can be employed for IR task. For example, the goal of *predictive coding* in E-Discovery is to develop some algorithms which can learn to assign either relevant or non-relevant label to a document (Cormack and Grossman, 2014). In order to be able to do this, the algorithm is given a *seed set*, which we can regard as examples of documents that have been humanly assigned some relevance labels. The algorithm then learns patterns from the *seed set* which it uses for onward classification.

This is purely a classification task, and the decision is either relevant (*R*) or not relevant (*NR*) (i.e., 2-class classification). Assigning binary labels to document is trivial, and linear algorithms like the Support Vector Machine (SVM) (Joachims, 1998) and Random Forest (RF) (Liaw and Wiener, 2002) have proven effective in text classification or categorization tasks (Sebastiani, 2002). Clustering, for instance by a centroid approach like the K-Means (Hartigan and Wong, 1979) or topic models such as LSA (Deerwester et al., 1990) and LDA (Blei, Ng, and Jordan, 2003) have also been effective in this regard.

In (Rohan et al., 2017), we employed a combination of clustering approaches for our system at COLIEE 2017 Legal information retrieval task. Obviously, for a set of documents that are relevant to a query, a user would most likely want to have the documents with label *R* to be ordered according to their relevance. When a binary classification task is formalized as a ranking problem, such that the goal is not only to assign relevance labels *R* or *NR* but to also rank in the order of their relevance, it is called *learning to rank* (L2R) (Li, 2011; Li, 2014).

The Learning-to-Rank task can be formalized as follows. Given the training set Q = {q$_1$, q$_2$, ..., q$_m$}, D and Y = {1, 2, ..., l} which are the sets of a query, document, and label respectively. Assuming the label is graded such that $l \Rightarrow l-1 \Rightarrow .... \Rightarrow 1$, where $\Rightarrow$ is used to denote the degrading order relation. Suppose there exist D$_i$ $\in$ D such that D$_i$ = {d$_{i,1}$, d$_{i,2}$, ..., d$_{i,n_i}$}, and q$_i$ is the i-th query corresponding to the set of documents in D$_i$ with labels y$_i$ = {y$_{i,1}$, y$_{i,2}$, ..., y$_{i,n_i}$}. Here n$_i$ denotes the sizes of D$_i$ and y$_i$; d$_{i,j}$ is the j-th document in D$_i$ and y$_{i,j}$ $\in$ Y is the j-th grade label y$_i$ which shows how relevant d$_{i,j}$ is to query q$_i$. The training set is represented as a tuple S = {(q$_i$, D$_i$), y$_i$ }$_{i=1}^{m}$.

If we represent each query-document pair with a feature vector, such that x$_{i,j}$ = ϕ (q$_i$, d$_{i,j}$), where i = 1, 2, ..., m; and j = 1, 2, ..., n$_i$ where ϕ is a function for translating each pair (q$_i$, d$_{i,j}$) into a feature vector. For each q$_i$, we can represent the feature vector with all its corresponding documents as x$_i$ = {x$_{i,1}$, x$_{i,2}$, x$_{i,3}$, ..., x$_{i,n_i}$}. We can represent the transformed dataset as S$'$ = {(x$_i$, y$_i$)}$_{i=1}^{m}$, where x $\in$ χ and $\chi \in \mathbb{R}^d$. The goal is to construct a model F(q, D) = F(x) that assigns a relevance score to each element of x. Liu (Liu, 2009) specified three categories of L2R based on the training objective, i.e., the *pointwise*, *pairwise*, and the listwise approaches (Liu, 2009). Also, different input features may be used in these models. A prominent pairwise loss function is the RankNet (Burges et al., 2005), and LambdaRank (Burges, 2010) for listwise training objective. Our training objective in the *learning to rank* task is a kind of listwise objective loss function.

As discussed earlier, most of the rank-based models discussed in section (2.6), especially the probabilistic models like BM25 can be used to assign relevance score. As we will see in chapter (6), for the ad-hoc retrieval task of E-Discovery, we employ a neural network which assigns a relevance score based on the feature vector obtained by encoding each word in the query and the associated document with some embedding features.

Our NN is an ensemble model based on Siamese architecture, with each component of the model obtaining a representation of either the document or the query. The benefit of our approach is that we are able to obtain a high-level semantic representation of documents and queries, which allows for a semantic matching. Subsequently, the network also learns to rank by feeding it positive relevant documents and negative irrelevant documents as sample.

Moreover, a Neural Network is a simplification of the human brain. Neural Networks can be seen as computational models based on a parallel processing which is able to adapt themselves to a specific task or learn from some data and generalize their outputs on an unseen data. Our brains have billions of connected neurons sharing signals amongst each other. Similarly, a Neural Network consists of a layered interconnected set of nodes which communicate by sending signals over a number of weighted connections (Zurada, 1992). Here, the lower layer receives some inputs, performs some computation on the input and passes its output to the layers above it. The bare-bone of every Neural Network is the Perceptron (Rosenblatt, 1958), as shown in figure 2.5. The Perceptron is computed by equations (2.38) to (2.39).

$$Z = \sum_i w_i x_i \tag{2.38}$$

$$y = f_N(Z) \tag{2.39}$$

where w$_i$ is the weight assigned to an input x$_i$, z is the node (summation) output and the function f$_N$ is a nonlinear function which produces the perceptron output y. A Neural Network is composed of an input layer where the inputs are received, one or more hidden layers where the interconnected nodes perform some computation to generate some high-level representation of the input, and an output layer. The property of the output layer depends on the task at hand, for example, in a 2-way classification task, the output layer is as shown in Figure 2.6. Neural Networks are especially powerful because of their non-linearity, i.e., they can obtain a good classifier on a model with non-linearly separable inputs. A simple feed-forward Neural Network with fully connected layers is

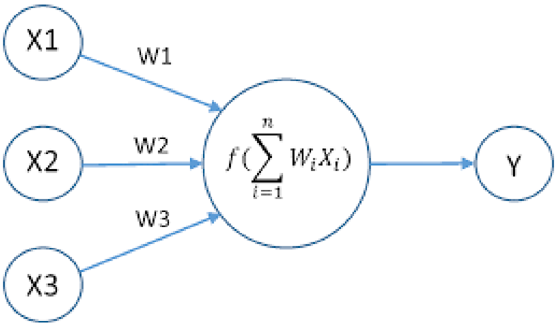


FIGURE 2.5: The Perceptron Neural Network

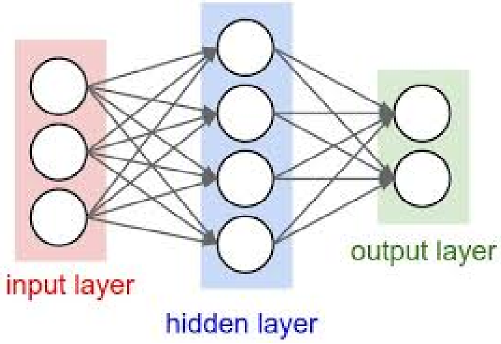


FIGURE 2.6: A simple 2-way classification Network with one hidden layer

represented by the equation (2.40) below:

$$\vec{y} = \tanh(W_2 \cdot \tanh(W_1 \cdot \vec{x} + \vec{b_1}) + \vec{b_2}) \tag{2.40}$$

where W_1, W_2, b_1, b_2 are the weight matrices and the bias vectors respectively, which are parameters to be learned in order to minimize the loss function. The *tanh*, the hyperbolic tangent is a non-linear function. Most networks are trained with the back-propagation algorithm (Bengio, 2009). A network with many hidden layers has more representational power, and a network with several hidden layers, most especially with a residual connection, is said to have more depth, hence, the connotation -*Deep Neural Network* (Schmidhuber, 2015).

2.11 Chapter Summary

In this chapter, we articulated the desirable features of an IR system, which is applicable across the tasks that the thesis describes. We enumerated different models of IR, starting from the Boolean to the probabilistic models. We discussed various approaches used to improve IR performance, these include query expansion, relevance feedback etc. We also introduced word embedding, the use of which is central to many solutions described in this thesis. We also introduced the learning to rank with the aim of showcasing how the ad-hoc retrieval problem may be visualized as a machine learning problem. Overall, the chapter gives a basic understanding and important terminologies needed to make the succeeding chapters self-contained and understandable. In the next chapter, we discuss our work on *passage retrieval*, that is, a system that retrieves a relevant portion of a document in response to a query. As we will see, the queries are expanded concepts from an ontology. We also describe our method to segment document into topical units for this kind of retrieval. The solutions that we describe in that chapter would give critical exposition to understanding the later chapters.

Chapter 3

Document Segmentation for Fine-grained Information Retrieval

Given the peculiar nature and size of legal documents, in chapter 1, we identified the issue of *granularity* as one of the three challenging issues that a LIR has to overcome. Specifically, most retrieval systems tend to retrieve whole documents, which constitutes a problem of *information overload*. In this chapter, we motivate a passage retrieval solution that works at the level of document units which we refer to as segments. The bone of contention has often been what constitutes a document unit/segment, e.g., is it a sentence or a paragraph, a fixed number of sentences or paragraphs, or some structured sections of a document, especially since some legal documents have sectionalized structure. Our proposal adopts a natural language processing solution which divides a document into coherent topical sections. The approach is intuitive since, in practice, a user would want to retrieve a passage whose sentences have a thematic alliance. This chapter gives a detailed background required to appreciate the proposed solution. We also introduced the relevant components of the overall system. These components, i.e., the text segmentation unit, and the text similarity unit are essential for the functioning of the proposed system. In chapter 5, we give a description of the main system that incorporates these sub-components, as well as the result obtained from the experiments that we ran.

In particular, the contribution that we present here includes the following:

- A novel text segmentation algorithm based on topic modeling and entity coherence.
- A novel semantic annotation framework for mapping legal text segments with controlled concepts, which combines approaches that induce knowledge from distributional word embedding and large-scale encyclopedic data like the Wikipedia.
- An approach to reducing information overload during retrieval activities.

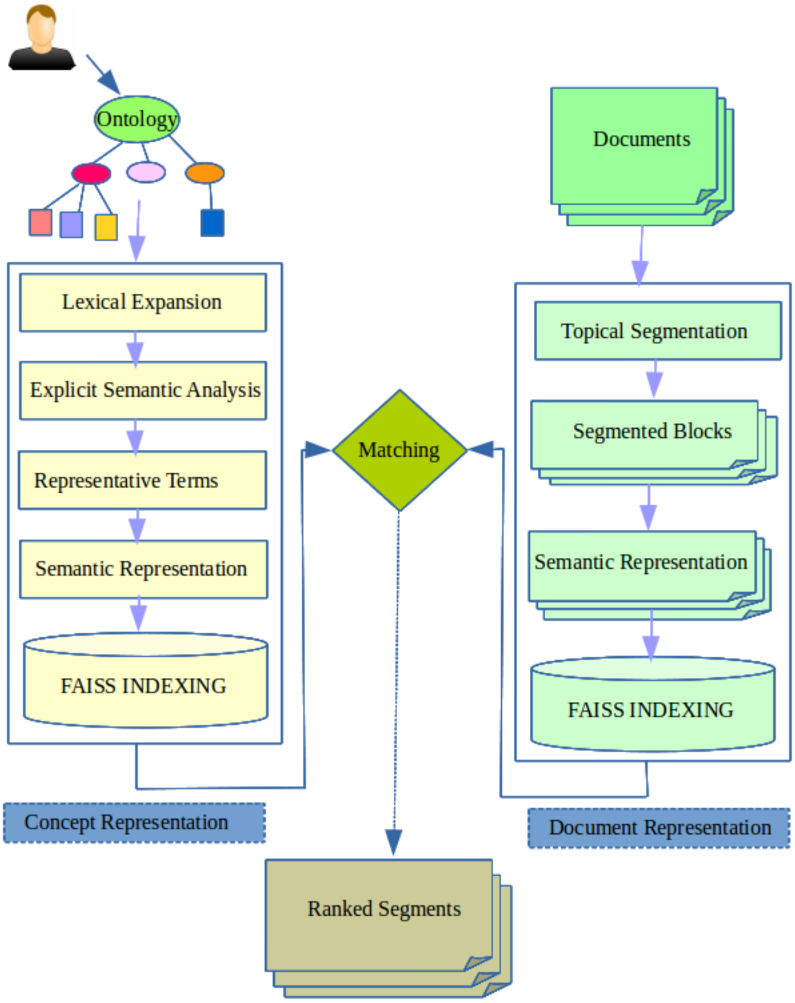


FIGURE 3.1: Architecture of the Conceptual Passage Retrieval System

3.1 Justifying the Conceptual Passage Retrieval

The demands of citizen that there should be transparency, accountability, and openness in the dealings of government by making government data to be accessible to the public have in part contributed to the increasing number of legislative documents available on the Internet. The guiding principles of open data say that data must be complete, permanently available, timely, accessible, documented and safe to open[1]. In the past, private legal information systems like Lexis Nexis, and Westlaw have maintained the monopoly of providing access to supreme and federal court cases, however, we now have websites like EUR-Lex, PublicData[2] and Europa[3] which contain millions of archived legislative documents that are of concern to the European Union. Generally, the documents usually archived can be categorized into three, i.e., normative documents such as decrees and acts; preparatory works which are products of legislative processes, and lastly, court

[1] https://opengovdata.org/
[2] http://publicdata.eu/
[3] https://data.europa.eu/euodp/en/home/

judgments which show how rules are being interpreted (Lyytikainen, Tiitinen, and Salminen, 2000). The websites are also frequently updated as new documents arrive. Case laws are particularly important for they are records of the proceedings of a court, and play important roles in precedence search in which legal rules from past judgments can be assimilated to prepare arguments for a similar case. Since it is the stock-in-trade of legal practitioners and lawyers to do extensive legal research while preparing their arguments, these massive resources that are freely available on the internet are of immense importance.

Most of the websites offering open legislative data offer different document search criteria, for example, since most documents come with meta-data, it is possible to search based on attributes like publication date, document origin, document type etc. The XML, as an important component of the semantic web standard is a markup language which uses some rules to encode documents such that machines can better read and make much sense from the document. The importance and use of XML in legislative documents has been well reported in the literature (Palmirani and Vitali, 2012). The set of rules (i.e., the lexicon, syntax, and grammar) and the tags it uses are also customizable for any specific domain (Boella et al., 2016). XML also provides some meta-rules or structure which may be used as meta-data for querying a database. One may argue about the role of legislative XML standards, e.g., XML standards with national jurisdiction like the Italian NormaIn-Rete, Danish Lex-Dania, Swiss CHLexML, Brazilian LexML and the Austrian eLaw or the more continental frameworks like the European Metalex interchange formats and the Akoma Ntoso (Palmirani and Vitali, 2011) which has been specially designed for African legislative documents. However, the reality is that each legislative text comes with its distinctive characteristics. Also, the fact that XML may help with management and retrieval of norms does not translate to a capability to offer information about the semantics of a document (Boella et al., 2016). Furthermore, the meta-data and how they are used for classification varies from one document type to another or one website to another, causing a lot of inconsistencies which pose serious problems for users since it is difficult to formulate the query that will isolate a specific document.

3.2 Ontology and Legal Document Modeling

In order to provide a unified standard, concepts from ontologies have been used to index documents from these websites. An ontology is defined as a formal conceptualization of the world, capturing consensual knowledge in a specific domain (Kiyavitskaya et al., 2006). As Boella et. al. (Boella et al., 2016) notes, anthropological and psycholinguistic studies support the intuitive design of ontologies as a way to modeling the relations between concepts. This is done through a hierarchical listing of a detailed category of a concept into a more specific category of the concept. Hence, they offer a way to performing a semantic analysis of the document. Practitioners in the legal domain tends to perceive concepts in a normative way. Existing ontologies in the legal domain include the

LOIS project (Schweighofer and Liebwald, 2007) which was developed based on DOLCE (Gangemi et al., 2002), the ONTOMEDIA project (Fernandez-Barrera and Casanovas, 2011) as well as the Legal Taxonomy Syllabus (Ajani et al., 2007) which has been incorporated into the EUNOMOS (Boella et al., 2016) legal document management system.

In particular, concepts from Eurovoc thesaurus have been widely used to index EU publications that are available on most public databases like the EUR-Lex. An ontology-based efficient retrieval of legal resources is possible by allowing users to query the database based on conceptual terms as opposed to ordinary keyword search, or the grossly inefficient Boolean search which also defaults to ’exact’ keyword search. Here, we do not concern ourselves with whether or how a text is marked-up with any metadata for our technique as well as the texts in our dataset have none of such markups.

Figure 3.1 shows the general architecture of our proposed conceptual passage retrieval system. The problem we try to solve is what projects like EUNOMOS (Boella et al., 2016) and EULEGIS (Lyytikainen, Tiitinen, and Salminen, 2000) slightly overlook, i.e., that of granularity of retrieval, such that the problem of information overload is adequately taken care of. On the average, legal documents are usually long and a user may only be interested in a particular part(s) of a document instead of the whole document. A system that is able to retrieve specific portion(s) of a document that is of interest to a user would definitely be appealing. In addition, such a system can reduce the process of manual filtering which users would otherwise go through in search of relevant passages in a text. These kinds of IR systems are referred to as *passage retrieval* systems. The benefit of a passage retrieval system cannot be overstated, for example, the precision with which a retrieval system will map a query to a section containing ten sentences will be much higher than that of a full document containing 20 pages covering different subjects or topics (Salton, Allan, and Buckley, 1993; Callan, 1994). The work of Tellex et. al. (Tellex et al., 2003) was one of the earliest passage-based question answering system. The authors in (Rosso, Correa, and Buscaldi, 2011) describes their experiments with the JIRS system on a range of passage retrieval tasks using patent documents. As we will see, our language processing techniques clearly differs from these systems; more importantly, our system incorporates the use of domain knowledge, which we formalize as a semantic annotation task.

The question to be asked is what constitutes an acceptable section of a document? is it a fixed number of sentences? is it a paragraph or fixed number of paragraphs? or is it a formatted XML section? Most legal documents are formatted using markups, this means that they are already highly structured, mostly into partial sections (Moens, 2001), nevertheless, discourses in the sections are still unstructured text . A keen look at the structure would reveal that even a section may contain other sub-sections. Moreover, each section or sub-section may still be several pages long, thus containing many details on a diversity of subjects. A solution is to group contiguous sentences that talks about the same topic into the same section[4]. A document normally contains a mixture of topics. Therefore, if topics and subtopics in a document are identified, such that the coherent ones form separate groups, then it would be much easier to associate concepts with these topical groups in a way to improving and making retrieval more efficient.

In a sense, the main task here can be divided into two subtasks. The first subtask is to divide a document into topical group which shares the same semantics. The second task is to obtain a semantic representation for each concept as well as the topical group, and then determine if the representation of a concept matches that of a segment. The second task is defined in this thesis as semantic annotation task.

3.3 Segmenting Document By Topics

3.3.1 Text Segmentation

The goal of Text Segmentation (TS) is to identify boundaries of a topic shift in a document. As previously highlighted, discourse structure studies have shown that a document is made up of topics and sub-topics exhibited by its constituent units e.g., words, sentences and paragraphs. The dimension of a shift in topics is, therefore, a function of the semantic bond and relationship within these units. Intuitively, the bond tends to be higher among units with common topics. This notion is what is termed *cohesion* or *coherence* within a document. Cohesion is a function of grammatical factors, e.g., co-reference and sentential connectives as well as lexical factors like collocation (Kaufmann, 1999). It is, therefore, possible to identify the point in the document where there is a change in topic by monitoring the changes in the ways words are used in the document (Halliday and Hasan, 2014). Obviously, it makes sense to assume that document units with a similar topic would have many words in common. The process of dividing a text into portions of different topical themes is called Text Segmentation (Hearst, 1997).

The text units (*sentences* or *paragraphs*) making up a segment have to be coherent, i.e., exhibiting strong grammatical, lexical and semantic cohesion (Kaufmann, 1999). Furthermore, such document units have to be contiguous, i.e., share the same context. Segmentation in the legal document is not new, however, the task is mostly done manually by experts, which is both laborious and expensive (Moens, 2001). Furthermore, the manually segmented sections may not be entirely fine-grained. It is therefore important to design algorithms that are able to automatically model language synthesis and define sections in the document. As we will see in the later part of this chapter, Our goal is to automatically obtain topical segments of any legislative document. The proposed approach is an unsupervised method which relies on topics obtained from LDA topic modeling of some documents. Furthermore, we incorporate entity coherence (Barzilay and Lapata, 2008),

[4] Throughout this chapter, we interchangeably use the terms section, block or segment as referring to the same thing

that allows the introduction of some heuristic rules for boundary decision. Once the segments are obtained, they are used as inputs to the semantic annotation module which performs semantic analysis of each segment and associates an appropriate concept to the segment.

3.3.2 Approaches To Text Segmentation

Available Text Segmentation systems can be categorized into two broad groups, i.e., Linear and Hierarchical Text Segmentation systems. The most popular ones are the Linear TS algorithms (Choi, 2000; Hearst, 1997; Beeferman, Berger, and Lafferty, 1999). Linear TS algorithms observe a sequence of topic shift without considering the sub-topic structures within segments. On the other hand, hierarchical TS algorithms (Eisenstein, 2009) are more fine-grained, for it is possible to visualize even the minutest detail about the sub-topic structure of a document. Most of the published work have relied on the use of similarity in vocabulary usage in sentences in order to detect potential topic shift (Hearst, 1997; Choi, 2000). The lexical relationship that exists between some contiguous text units is used as a measure of coherence. These lexical relationships include vocabulary overlap which could be identified by word stem repetition, context vectors, entity repetition, word frequency model and word similarity (Hearst, 1993; Kaufmann, 1999; Beeferman, Berger, and Lafferty, 1999; Reynar, 1999; Utiyama and Isahara, 2001). High vocabulary overlap between two compared units is taken to mean high coherence and vice versa. This idea, otherwise known as *lexical cohesion* has the disadvantage of failing due to *lexical ambiguity*. The TextTiling algorithm (Hearst, 1997) is a typical example of TS systems in this category. TextTiling works by assigning a score to each topic boundary candidate within k chosen window. Topic boundaries are placed at the locations of valleys in this measure and are then adjusted to coincide with known paragraph boundaries. The authors in (Choi, Wiemer-Hastings, and Moore, 2001) build on this idea with the introduction of a similarity matrix neighborhood ranking, where the rank of an element corresponds to the number of neighbours with lower values.

We discussed in the early chapters how ambiguity (as expressed by synonymy and polysemy) poses a big problem in natural language processing. For instance, when orthographically different but synonymous words are used within the units of a document, lexical cohesion-based algorithms are unable to group such units as a segment. A natural solution is to incorporate approaches that overcome ambiguity problems. Researchers then proposed the use of topics (Choi, Wiemer-Hastings, and Moore, 2001; Riedl and Biemann, 2012b; Du, Pate, and Johnson, 2015; Dias, Alves, and Lopes, 2007). These works are mainly inspired by distributional semantics-based approaches such as the LSA (Landauer, Foltz, and Laham, 1998; Choi, Wiemer-Hastings, and Moore, 2001) and Latent Dirichlet Allocation (LDA) topic models (Riedl and Biemann, 2012b; Misra et al., 2011). The second approach which is mostly used is the discourse-based techniques. This approach relies on the use of cue phrases and Prosodic features, e.g., pause duration that is most probable to occur close to a segment boundary. These features are combined using a machine learning model (Beeferman, Berger, and Lafferty, 1999; Passonneau and Litman, 1997; Reynar, 1999). This approach, however, is domain independent and can only perform well if the system is evaluated on documents which use the same cue words.

Recent work (Du, Pate, and Johnson, 2015; Misra et al., 2011; Riedl and Biemann, 2012b) employed topic modeling techniques using algorithms like LDA (Blei, Ng, and Jordan, 2003). The idea is to induce the semantic relationship between words and to use frequency of topic assigned to words by LDA instead of the word itself to build sentence vector. This makes sense since a word could appear under different topics thus partially overcoming lexical ambiguity. Our proposed approach builds on the previously published work by employing a topic modeling algorithm to reveal the topical structure of any given document. Furthermore, we introduce two heuristics, i.e., (*lexical* and *semantic*) heuristics which are used solely for boundary adjustment. For instance, a position *m+1* after a sentence S_m is a valid boundary only if sentences within the region S_{m-k} and S_{m+k} have no common entities, where *k* is a chosen window. Also, coherent sentences tend to have similar semantics. This is the main idea in TextTiling and Choi's work Hearst, 1993; Choi, 2000 with the exception that they rely on term frequency to build sentence vector used for similarity calculation. Since this approach suffers from lexical ambiguity, e.g. the word *dog* appearing in one sentence followed by *puppy* in another is not deemed to be similar, we incorporate a semantic-net based similarity using WordNet. This typically overcomes the *synonymy* problem for a more efficient similarity calculation. The two heuristics were combined in a way to help in boundary decision making with topic-based sentence similarity. The approach can be summarized into the following steps:

1. Obtain the topic model of a sample corpus by modeling with LDA algorithm.
2. Tokenize each input document into sentences
3. Obtain the topics of each sentence using the topic model in step 1
4. Obtain the topical similarity of sentences and cluster the contiguously similar ones
5. Validate contiguity with a WordNet-based sentence similarity calculation
6. Perform boundary adjustment using Entity Coherence

We now proceed to explain these steps in detail.

3.3.3 Topic Segmentation With LDA

Given an input document **W**, our algorithm divides the document into a set of minimal text units (s_1, s_2, s_3, ..., s_T), where T is the number of sentences in the document, each s_i can be viewed as a pseudo-document that contains a list of tokens v ∈ V, where V is the set of vocabulary of *W*. In practice, the goal is to identify sets of contiguous s_i that are mono-thematic, each member of the set being a segment. Following similar work

(Du, Pate, and Johnson, 2015; Misra et al., 2011), we also employed LDA topic modeling algorithm (Blei, Ng, and Jordan, 2003; Blei and Lafferty, 2006) to obtain topics for each word. Moreover, topic models are a suite of an unsupervised algorithm that uncovers the hidden thematic structure in a document collection. Modeling documents based on topics provide a simple way to analyze a large volume of unlabeled text while exposing the hidden semantic relationship between them. The LDA algorithm is briefly described in section (3.3.4).

3.3.4 The LDA Algorithm

LDA is a generative probabilistic model of a corpus with the intuition that a document is a random distribution over latent topics, where each topic is characterized by a distribution over the words in the vocabulary. Say for instance that a document is perceived as a bag of words where the order does not matter, suppose that the fixed number of topics (say for instance n_T) is known. Considering there could be many of such documents in a bag, then each word in the bag is randomly assigned a topic t drawn from the Dirichlet distribution. This gives a topic representations of the documents and the word distribution of all the topics. The goal is then to find the proportion of the words in document **W** that are currently assigned to each topic t as well as the proportion of assignments to topic t over all documents that come from this word w. In other words, a Dirichlet distribution of each word over each topic is obtained. The model has shown capability to capture semantic information from documents in a way similar to probabilistic latent semantic analysis (Hofmann, 1999). The idea is to induce a low dimensionality representation of the text in the semantic space while preserving the latent statistical features of each text.

Formally, given a document **w** of N words such that **w** = (w_1,w_2,w_3...w_N) and a corpus D of M documents denoted by D = ($\mathbf{w}_1$,$\mathbf{w}_2$,$\mathbf{w}_3$.....$\mathbf{w}_M$). For each of the words w_n in the document, a topic z_n is drawn from the topic distribution θ, and a word w_n is randomly chosen from P(w_n | z_n, β) conditioned on z_n. Given α, a k-vector with components with $\alpha_i > 0$ and the Gamma function Γ(x). The probability density of the Dirichlet is given as

$$P(\Theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\Pi_{i=1}^{k}\Gamma(\alpha_i)} \Theta_1^{\alpha_1 - 1} ....\Theta_k^{\alpha_k - 1} \tag{3.1}$$

Given the parameters α and β, the joint distribution of a topic mixture θ, a set of N topics **z**, and a set of N words **w** is thus given by

$$P(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) = P(\theta|\alpha)\Pi_{n=1}^{N} P(z_n|\theta)P(w_n|z_n, \beta) \tag{3.2}$$

Integrating over θ and summing of z, the set of topic assignments, the distribution of a document can be obtained as below

$$P(\mathbf{w}|\alpha, \beta) = \int P(\theta|\alpha) \left(\Pi_{n=1}^{N} \sum_{z_n} P(z_n|\theta)P(w_n|z_n, \beta) \right) \mathrm{d}\theta \tag{3.3}$$

where P(z$_n$ | θ) is θ_i for the unique i such that $z_n^i = 1$. The probability that a corpus is obtained through the product of marginal probability given in equation (3.3), for each $\mathbf{w}_n$ in D is given in equation (3.4):

$$P(\mathbf{w}|\alpha,\beta) = \left\{\Pi_{d=1}^{M} \int P(\theta_d|\alpha) \left(\Pi_{n=1}^{N_d} \sum_{z_dn} P(z_dn|\theta_d)P(w_dn|z_dn,\beta)\right) \mathrm{d}\theta_d\right\} \tag{3.4}$$

Training the LDA model on a corpus requires feeding the model with the set of tokens from the document. The model statistically estimates the topic distribution θ_d for each document as well as the word distribution in each topic. A model can also be used to predict the topic classes for a previously unseen document. In our work, we have trained the LDA algorithm on different datasets, these include the JRC corpus[5] which is a collection of legislative documents, Wikipedia dump[6], and lastly the Choi's dataset[7] (Choi, 2000).

3.3.5 Computing Sentence Similarity with LDA

Riedl and Biemann in (Riedl and Biemann, 2012a) utilized the most frequent topic assigned to a word after the Gibbs inference in order to avoid the instability that is usually associated with a generative algorithm like the LDA. Contrarily, for each sentence, we obtain the distribution of topics for each word along with their probability score. Next, we select the topic with the highest probability for each word. For each sentence, this results into a bag of topics where order does not matter. This can be seen as a matrix $G = L \times T$ where $l \in L$ is a vector of length *k*, the chosen number of topics. Each vector l contains the frequency of each topic ID assigned by the LDA to the words in a sentence, where, by topic ID, we denote the topic group or cluster that a word belongs, i.e., a number in the range [0, $T - 1$]. As an example, assuming the number of topics n = 10 and the bag of topics for a sentence is $\{0, 0, 5, 2, 3, 3, 7, 7, 1, 6, 5\}$, then the vector for such a sentence will be [2,1,1,2,0,2,1,2,0,0], each element representing the frequency of occurrence of topics 0 to 9. A general assumption is that sentences with similar topics have some semantic relationship. Furthermore, the LDA is able to unravel the latent relationship between words through its probabilistic clustering.

We introduce a parameter, w_n, called the lookahead window. This works similarly to the *k-block* of sentences employed in (Riedl and Biemann, 2012b) but with a different objective. The previous work compares the vector of a sentence to the *k-block* of sentences on the left and the right of a sentence in order to get the similarity score[8] for that sentence. The process is then repeated for each sentence in the document in order to calculate its

[5] Available at https://ec.europa.eu/jrc/en/language-technologies/jrc-acquis

[6] The wikipedia dump was downloaded on July 30, 2015. It is accessible at https://dumps.wikimedia.org/enwiki/.

[7] Available at http://web.archive.org/web/20040810103924/http://www.cs.man.ac.uk/~mary/choif/software.html

[8] Otherwise called coherence score.

similarity to the surrounding sentences. In our implementation, for each pass over the list of sentences, using the lookahead window[9], we sum up the vectors of sentences within the window and use it as a *reference* vector for sentences within that window. The intuition is that we can treat the set of sentences within a window as a *mini*-document, summing up the vectors give the overall representation of the *mini*-document. It is therefore, possible to estimate the semantic distance between the *mini*-document and each neighour sentence. Sentences with a high topic correlation will have a high similarity to the reference vector. Figure 3.2 shows the process of summing over the vector for a sample document of 10 sentences. Once the reference values have been obtained, the next step is to obtain sentence similarity, otherwise called the coherence score. To do this, for each window, we use the cosine similarity between each sentence and the reference vector. Repeating this for all the sentences results into a time series, e.g., a one-dimensional vector of similarity values over all the sentences.

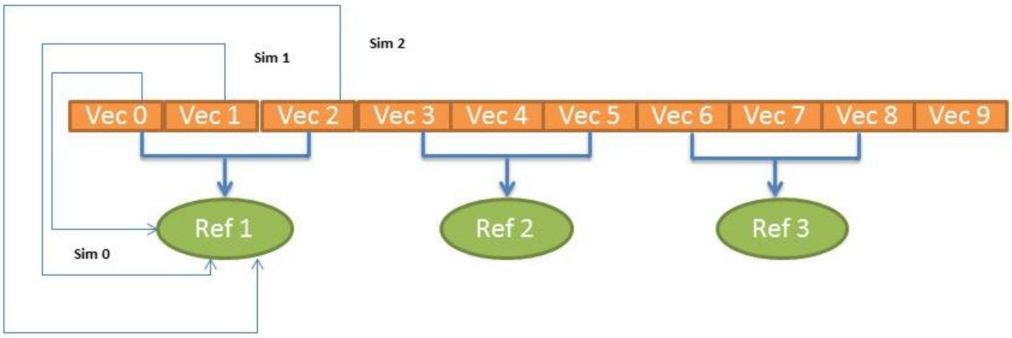


FIGURE 3.2: Summing over window vector

3.3.6 Feature-Enriched Classifier-based Sentence Similarity

This section provides a validation for the sentence similarity calculation performed in the previous step. This is achieved by incorporating an extra sentence similarity verification procedure. The similarity calculation is as done in the preceding section except that we introduced lexical and semantic similarity calculation method. In particular, our approach is to extract some descriptive features from the text which a machine learning classifier aggregates and learns in order to measure how similar two text snippets are. We trained a Support Vector Machine (SVM) (Chang and Lin, 2011) classifier. The input to the classifier is the extracted features i.e., the lexical features like the word ordering and word overlap similarity and a semantic similarity feature with WordNet. Below, we describe the important features used by the classifier to compute similarity.

[9] From our observation, we found out that the best default value is w_n = 3.

Word Ordering Feature

We use the union of all tokens in a pair of a sentence to build a vocabulary of non-repeating terms. For each sentence, the position mapping of each word in the vocabulary is used to build a vector. In order to obtain the position mapping, a unique index number is assigned to each vocabulary term. Similarly, in order to obtain the word order vector of a sentence, each term in the vocabulary is compared against the terms in the sentence. If a vocabulary term is found in the sentence, the index number of that term in the vocabulary is added to the vector. Otherwise, a similarity of the vocabulary terms and each term in the sentence is calculated using a *WordNet*-based word similarity algorithms (Adebayo, Di Caro, and Boella, 2016c). The index number of the sentence term with the highest similarity score above a threshold is added. If the first two conditions do not hold, a score, 0, is added to the vector. Consider two sentences S1 and S2,

S1: A panda bear

S2: A baby panda

Then the vocabulary is a list that contains the union of tokens in *S1* and *S2* as shown below:

Vocabulary = A, baby, bear, panda

Vocabulary-Index = A:1, baby:2, bear:3, panda:4

and the sentences are transformed to the vectors below:

S1 = 1,0,3,4

S2 = 1,2,4,4

In the example, the vocabulary term *bear* does not exist in *S2*. Obviously, the term *'bear'* is more similar to the term *'panda'* than all the terms in *S2*. The index number of *panda* is thus assigned in place of *bear*. In *S1*, the vocabulary term *baby* is not similar to any term, thus 0 is assigned. The word ordering feature is then computed as the cosine of the vectors after the WordNet-based similarity transformation.

Word Overlap Feature

We use the word n-gram overlap features of (Saric et al., 2012). The n-grams overlap is defined as the harmonic mean of the degree of mappings between the first and second sentence and vice versa, requiring an exact string match of n-grams in the two sentences.

$$\text{Ng(A,B)} = \left(2(\frac{\mid A \mid}{A \cap B} + \frac{\mid B \mid}{A \cap B})^{-1}\right) \tag{3.5}$$

Where A and B are the set of n-grams in the two sentences. We computed three separate features using equation 2 for each of the following character n-grams: Unigram, Bigram, and Trigram. Furthermore, we include the weighted word overlap which uses information content (Resnik, 1995).

$$\text{wwo(A,B)} = \frac{\sum_{w \in A \cap B} ic(w)}{\sum_{w' \in B} ic(w')} \tag{3.6}$$

$$\text{ic(w)} = \left(\ln \frac{\sum_{w' \in C} freq(w')}{freq(w)} \right) \tag{3.7}$$

Where C is the set of words and *freq(w)* is the occurrence count obtained from the Brown corpus. Our weighted word overlap feature is computed as the harmonic mean of the functions *wwo(A,B)* and *wwo(B,A)*.

Word-to-Word WordNet Similarity Feature

In order To compute similarity between two sentences using the WordNet, it is possible to calculate how similar each word in the first sentence is to the words in the second sentence. When the similarity scores are aggregated, we may have an idea of how similar thee two sentences are. Usually, there are existing techniques for computing the similarity between two words using any thesaurus like the WordNet, e.g., by using the path length between two words in a taxonomy (Resnik, 1995). However, as pointed out by (Li et al., 2006), this obviates the distance knowledge that can be easily observed from the hierarchical organization of concepts in most semantic nets. As a solution, the depth function was introduced, with the intuition that the words at the upper layer of a Semantic Net contain general semantics and less similarity, while those at lower layers are more similar. Therefore, the similarity should be a function of both the depth and the path length distances between two concepts. Here, we use both the path length between each word as well as the depth function. Usually, a longer path length between two concepts signifies a lower similarity. If f1(h) is a function of the depth and f2(l) is a function of the length, then the similarity between two words is given by:

$$S(w_1, w_2) = f1(h).f2(l) \tag{3.8}$$

The length function is a monotonically decreasing function with respect to the path length l between two concepts. This is captured by introducing a constant alpha.

$$\text{f2(l)} = e^{-\propto l} \tag{3.9}$$

Likewise, the depth function is monotonically increasing with respect to the depth h of concept in the hierarchy.

$$\text{f1(h)} = \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} \tag{3.10}$$

The similarity between two concepts is then calculated by:

$$S(w_1, w_2) = e^{-\propto l}.\frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} \tag{3.11}$$

Li et. al. (Li et al., 2006) empirically discovered that for optimal performance in WordNet, alpha should be set to 0.2 and Beta set to 0.45. We compare each word in the first sentence to each word in the second sentence, obtaining the similarity score. For each pair being compared, if the similarity score is less than < 0.25 then that similarity value is dropped. The final similarity is computed by summing the pair similarity values greater than 0.25 and dividing by the total count of these similarity scores. The similarity feature is obtained using the formula in equation (3.12), where the default threshold was fixed at 0.25.

$$\text{Sim} = \frac{\sum_{i,j}^{m,n} |S(w_i, w_j > x)|}{tCount} \tag{3.12}$$

Where *S*(w_i, w_j) is the similarity score for two words, *tCount* is the total number of the set of similarity scores that exceeds the threshold and *Sim* is the aggregating function combining all the pairwise similarities.

Embedding Similarity Feature

Using GloVe embeddings (Pennington, Socher, and Manning, 2014), the similarity between two sentences is computed as the cosine of the distance between the sentence embeddings. Assume that each sentence S contains words x$_i$, x$_{i+1}$, x$_{i+2}$, x$_{i+3}$, ..., x$_n$. We associate each word w in our vocabulary V with a vector representation x$_w \in \mathbb{R}^d$. Each x$_w$ is of dimension $d \times$ ∣V∣ of the word embedding matrix W$_e$, where ∣V∣ is the size of the vocabulary. For each sentence S, we generate an embedding representation by performing an element-wise sum of each x$_w \in$ S. We normalize the resulting vector sum by the length of the sequence as equation (3.13) shows.

$$S_{emb} = \frac{1}{|n|}\sum_{i=1}^{|n|} x_i, \quad S_{emb} \in \mathbb{R}^{d \times |V|} \tag{3.13}$$

where s$_{emb}$ denotes the embedding representation of a sentence.

Given a set of human annotated sentence-pairs along with their similarity scores which may be used as the training samples, our algorithm extracts the above features, the SVM algorithm then combines the features in order to learn the similarity. Next, we test the accuracy of the classifier using a set of sentence-pairs which the annotators have graded with the similarity score. Once the classifier achieves a reasonable accuracy level, we can put it to use to grade the similarity between any two given sentences.

Recall that our goal is to validate the similarity score obtained for two compared sentences when computed using the LDA as explained in section 3.3.5. Assume that we are to calculate the similarity between a given senntence A, and three other sentences B, C, and D and then rank the sentences according to the similarity score. First, we compute the similarity using the method described in section 3.3.5 and then using the SVM classifier. The good case is if the two methods report the same rank for the sentences. In the other case, a validation is needed and we simply utilize the rank or similarity scores computed by the SVM classifier as the correct similarity score. The main idea here is to detect boundary points in a text. This is done by computing the similarity of a sentence to its neighboring sentences. When a set of contiguous sentences are highly similar semantically, then we say that they belong to the same segment. Similarly, once the similarity drops sharply in-between two sets of contiguous but highly similar sentences, that signifies a break in topic, the end of a segment, or the beginning of another segment.

3.3.7 Entity-Based Coherence

Researchers working on discourse analysis have observed that the entity distribution and transition pattern in a text might be a good indicator of the points where there is coherence or topic shift in a text (Mann and Thompson, 1988; Grosz, Weinstein, and Joshi, 1995). The work of (Barzilay and Lapata, 2008) is based on the Centering theory, where the authors represent a document as a grid of entities in the document with their roles (subject, object, neither subject nor object, and absence) specified as the actions of these entities. The rows of the grid correspond to sentences, while the columns correspond to discourse entities. We adopt their idea in our work by observing the spread of entities across the sentences in the document to be segmented. Contrary to the grid-based entity ranking (Barzilay and Lapata, 2008), our goal is to observe entity overlap that exists between sentences within a chosen shift window[10]. Succinctly, we only use the information about entity coherence for the necessary boundary adjustment and not boundary detection to be specific. To achieve this, we use a grammar-based Regex parser to extract all the noun phrases in each sentence. There are existing tools for entity extraction, e.g., by using Stanford's Part-of-Speech (POS) tagger[11] to extract just the Nouns or any named entity recognizer (NER). However, this adds other overheads and dependencies, and also increases the time complexity of our algorithm. The Regex parser is simple, easily customizable, and not computationally effective. Moreover, we observed that it performed competitively to Stanford POS tagger. To determine the overlap for a sentence S$_i$, we compute the ratio of its common noun-phrases to its right neighbours within a specified window, e.g., { S_{i+1}, S_{i+2}, S_{i+3}}. The entity overlap is obtained as follows:

$$\text{EOV} = \frac{|A \,\tilde{\cap}\, B^*|}{|A \cup B^*|} \tag{3.14}$$

Where A and B^* represents the set of entities in the sentence being considered and right neighours within a specified window, respectively. The intersection, $\tilde{\cap}$, allows partial

[10] Following our previous lookahead parameter w_n, we use a window of 3 sentences as default.

[11] https://nlp.stanford.edu/software/tagger.shtml

matches since the entities are considered equivalent if there is an exact match or an entity is a substring of the other. Instead of using the overlap score, we record the last sentence from within the B^* that has shared entities with A if the overlap score actually exceeds a threshold. As an example, if a sentence S_1 is compared to $\{S_2, S_3, S_4\}$ with the entity overlap score between them exceeding the threshold, then, one by one, we check if it actually has an overlap with each of S_2, S_3 and S_4 independently. If say, for instance, we discover that S_1 and S_4 do not have any common entities but it has with S_2 and S_3, then the index of sentence S_3[12] is used as its last sentence collocation. It becomes plain whether a sentence share entities with its immediate neighbors. In this case, the assumption is that such a sentence is not likely to be a boundary. As an example, the text below shows how entity coherence may support boundary adjustment. The entities detected by our custom parser are in bold.

1. S_1: ***Cook*** *had discovered a* ***beef*** *in his possession a few days earlier and , when he could not show the* ***hide****, arrested him.*
2. S_2: *Thinking the evidence insufficient to get a conviction, he later released him.*
3. S_3: *Even while suffering the trip to his* ***home****,* ***Cook*** *swore to* ***Moore*** *and* ***Lane*** *that he would kill the* ***Indian****.*
4. S_4: *Three weeks later, following his recovery, armed with a* ***writ*** *issued by the* ***Catskill justice*** *on* ***affidavits*** *prepared by the* ***district attorney****,* ***Cook*** *and* ***Russell*** *rode to arrest* ***Martinez****.*
5. S_5: *Arriving at daybreak, they found* ***Julio*** *in his* ***corral*** *and demanded that he surrender.*
6. S_6: *Instead, he whirled and ran to his* ***house*** *for a* ***gun****, forcing them to kill him,* ***Cook*** *reported.*

In the example above, the entity *Cook* appears in S_1, S_3,S_4 and S_6. Considering S_1, we conclude that no boundary exists until S_4 since there is significant entity overlap with S_3 and S_4 when moving over the sentence window. Even though there appears to be no overlap with S_2 and S_1, it is safe to assume that S_2 is not a boundary since it falls within a coherent window, same goes for S_5 which falls within sentences S_3 and S_6. In our implementation, we create a vector whose elements hold the index of the last sentence it overlaps with. In case of no overlap, the entry for a sentence is set to 0. Identifying the entity distribution in this way is useful for a boundary adjustment for the suggested boundary from our topic based segmentation.

[12] We use index here to mean the unique ID of a sentence, e.g., sentence 1 will have index 0, sentence 2 will have index 1 etc..

3.3.8 Boundary Detection and Segmentation

As earlier explained, to detect boundaries of segments, it is important to focus on the points in a text where a sentence, or a few sentences suddenly appear to be less similar to a group of contiguous and highly similar sentences. We earlier described how to compute the similarity between the sentences in a text. The computed similarity scores, otherwise called coherence scores, are vectorized. To obtain the set of possible boundaries, we plot the coherence scores vector such that we can inspect the local Minima (valleys) and the local Maxima (peaks). The valleys are the smallest values within a local range of the coherence scores vector. Since coherence scores are higher within sentences sharing many topics, we assume that these points of minimum value signal the points where the least topic cohesion occurs, hence a segment boundary. The indices of the valleys [13] are collected in a vector as the potential points of a topic shift. We use the entries from the entity-based coherence described in the previous section to adjust the boundary. A mapping between the coherence vector and the entity-coherence vector is created. For each sentence in a document, each column of the entity coherence vector references the index of the last sentence it overlaps with. If there is a boundary after a sentence but there is an overlap reference to a sentence index higher than the boundary point then we *left-shift* the boundary as an adjustment task. Figure 3.3 shows the process of boundary adjustment over a sample sentence. In the example shown in the figure, possible segmentation has been obtained by the topic-based segmenter (see the break in columns of the first vector). We can see that for the first segment, the highest indexed entity overlap occurs at the sixth[14] sentence, which unfortunately belongs to another segment. Actually, both the fourth and fifth sentence of the first segment overlaps with the sixth sentence for the second segment. In this case, we shift this particular referenced sentence to the first segment for they cohere. This means that the first segment starts from the zeroth sentence and ends with the sixth sentence. The same thing applies for the third segment which has been adjusted. The idea here is based on Centering theory (Barzilay and Lapata, 2008), i.e., contiguous sentences with overlapping entities above a threshold exhibit coherence.

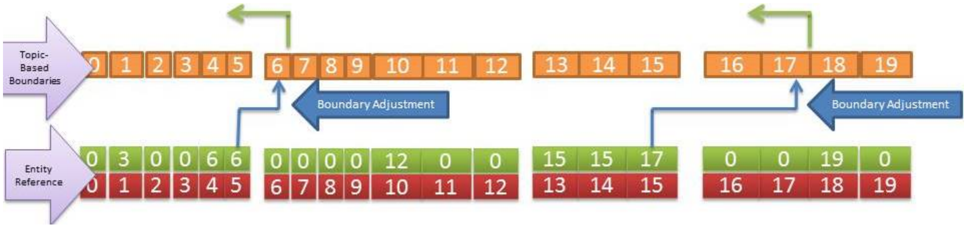


FIGURE 3.3: Entity Coherence-Based Boundary Adjustment

[13] i.e., the vector index which corresponds to the index of each sentence in the local minima

[14] considering the zero based ordering or indexing.

3.4 Associating Legal Concept(s) To Document Segments

Human processes information in terms of concept, therefore, it is ideal to have a system that allows its users to give concepts as queries to the system. In the legal domain, a commonly used ontology for conceptual indexing and cataloging is the EuroVoc thesaurus[15]. Along with the European Union publications office, EuroVoc is used by other European Union institutions, national and regional parliaments in Europe, as well as national administrations and private users around the world. Furthermore, EuroVoc is multilingual and multidisciplinary in nature. It contains concepts listed under twenty-one (21) main domains such as politics, European Union, Law, Finance etc, and the concepts are currently available in twenty-three (23) EU languages such as English, French, Italian, etc. Due to its broadness, It has been used by the European Union publications office for cataloging multilingual documents, for instance on EUR-Lex website. The advantage of tagging documents with concepts is obvious, for instance, users are able to navigate a document collection explicitly by using concept, therefore, providing a solution to the user specificity problem. Also, users can have an idea of the content or a preview of the content of the document because the concept label used for indexing is usually a descriptor of a broader knowledge (Pouliquen, Steinberger, and Ignat, 2006).

The important task is how to associate a concept label to the segment of a document that truly describes that concept. This requires devising a way to understand and represent the meaning of the concept as well as the meaning of each document/segment. It is possible to formalize the task as a Semantic Annotation problem. By Semantic Annotation (SA) (Bikakis et al., 2010), we refer to the process by which we map a concept to the specific document segment that it is most semantically related to. The rationale behind our framework is to provide legal practitioners and other end-users with an easy-to-use framework that allows for a fine-grained information retrieval. This works by providing a simple natural language processing tool that allows users to specify information need by using a controlled list of concepts, and the system retrieves not just the document related to the concept but specific part(s) of the document that is most semantically related to the concept.

The proposed system serves many purposes. First, users are freed from the rigours associated with query formulation. This is important because many people understand their information need, however, formulating the queries that represent such information need is cumbersome. By providing a controlled list of descriptors, such a problem is adequately taken care of. Secondly, from the perspective of information retrieval, concept mapping can support semantic query processing across disparate sources by expanding or rewriting the query using the corresponding information in multiple ontologies. The terms used in a document may be different from those expressed in an ontology (e.g., concept descriptors), that is, a concept descriptor being a generic term may not explicitly appear as a term in a document. The mapping process thus links the concept(s) to the

[15] http://eurovoc.europa.eu/drupal/

part of the document that most expresses its meaning. Conceptual retrieval emphasizes identifying and retrieving specific information that conforms to a specific retrieval concept. In other words, a fine-grained information retrieval can be achieved. Furthermore, an approach like this improves not only the precision but also the recall, which is an important metric for acceptability of any retrieval system in a domain-specific IR tasks like the E-Discovery (Socha and Gelbmann, 2005; Oard et al., 2010) in the legal domain which is generally classified as recall-oriented. Lastly, users are shielded from the problem of information overload since the system retrieves passages (or segments) which contain concise information that typically fits the concept selected for a query.

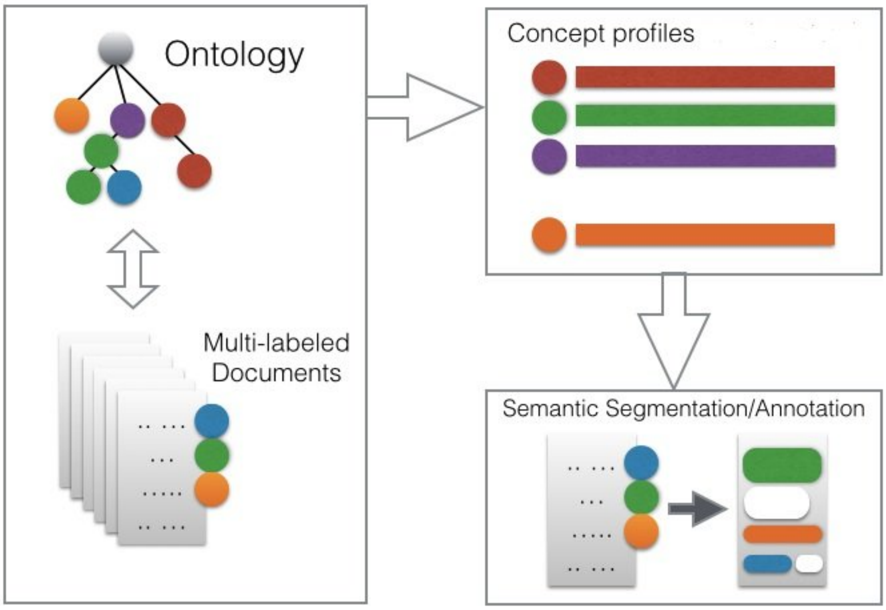


FIGURE 3.4: A Schematic Representation Of Semantic Annotation.

3.5 Semantic Annotation

Semantic Annotation (SA) is the process of mapping a chunk of a source text to distinct concepts defined by a domain expert. In other words, SA formalizes and structures a document with well-defined semantics specifically linked to a defined ontology (Popov et al., 2003). SA can be formalized as a 4-tuple {**Subj, Obj, Pred, Contx**}, where *Subj* is the subject of the annotation, *Obj* is the object of the annotation, *Pred* is the predicate which defines the type of relationship between *Subj* and *Obj*, while *Contx* signifies the context in which the annotation is made. As we can see, SA is a mapping function and can be used to add semantic information to the text. Figure 3.4 shows a pictorial representation of the task defined as semantic annotation in this thesis.

An Ontology is a formal conceptualization of the world, capturing consensual knowledge (Gruber, 1993; Kiyavitskaya et al., 2006). It lists the concepts along with their properties

and the relationship that exists between them. This study uses the Eurovoc [16] thesaurus as the ontology.
An ontology O := (C, $\leq_C$, R, $\leq_R$) is composed of four elements, i.e., (i) two disjoint set C (concept identifiers) and R (relation identifiers), (ii) a partial order $\leq_C$ on C which depicts the concept hierarchy, (iii) a function σ : R $\Longrightarrow$ C $\times$ C which is referred to as the signature and lastly, (iv) a partial order $\leq_R$ on R which is the relation hierarchy.
Similarly, we can derive a taxonomy/thesaurus from an ontology. The authors in (Dill et al., 2003b) defined a taxonomy as comprising of three elements:: a set of nodes V; a root r $\in V$; and a parent function, p: $V \mapsto V$. Where only the root is its own parent and serves as the ancestor of every node in the tree. Also, every other node is spawned from a parent node. Likewise, each node $v \in V$ is associated with a set of labels, $L(v)$. A node may also have siblings which are nodes from the same parent and have the same hierarchy level. We can incorporate information from the siblings in order to better disambiguate a concept for efficient annotation.

There are existing work on semantic annotation. GATE (Cunningham et al., 2002) is a semi-automatic annotation system based on NLP. GoNTogle (Bikakis et al., 2010) uses weighted k Nearest Neighbor (kNN) classifier for document annotation and retrieval. The authors in (Presutti, Draicchio, and Gangemi, 2012) developed a tool for ontology learning and population in the Semantic Web. Their approach utilizes Discourse Representation Theory and frame semantics for performing knowledge extraction. KIM (Popov et al., 2003) assigns semantic description to NEs in a text. The system is able to create hyperlinks to NEs in a text, indexing and document retrieval is then performed with the NEs. KIM uses a knowledge base called KIMO which contains over 200k entities. Furthermore, it relies on GATE for the NLP processing tasks. Regular Expressions (RE) have also been used to identify semantic elements in a text (Laclavik et al., 2006; Laclavik et al., 2007). It works by mapping part of a text related to semantic context and matching the subsequent sequence of characters to create an instance of the concept. Another named entity based annotation tool is GERBIL (Usbeck et al., 2015) which provides a rapid but extensive evaluation scheme for named entity recognition tools for the semantic web. Application of these systems includes document retrieval especially in the semantic web domain (Handschuh and Staab, 2002; Dill et al., 2003a). Eneldo and Johannes (Daelemans and Morik, 2008) performed semantic annotation on legal documents for document categorization. Using *Eurovoc* concept descriptors on *EurLex*[17], a ML classifier was trained for multi-label classification. Lawrence (Reeve Jr, 2006) employed SA for summarizing Biomedical texts based on concept frequency. Lawrence performed what he referred to as concept chaining, with an approach that mirrors the statistical concept clustering approach described in (Tegos, Karkaletsis, and Potamianos, 2008). These methods exploit the lexical and syntactic structure coupled with contextual information and dependencies between words for identifying relations between concepts. The work

[16] Eurovoc is available online at http://eurovoc.europa.eu/
[17] An online database of EU government documents. Available at http://eur-lex.europa.eu/

of (Kiyavitskaya et al., 2005) interfaces that of SemTag but relies on a grammar-based parsing to annotate entities such as email and web addresses, monetary formats, date and time formats, etc, with their respective concepts. SemTag (Dill et al., 2003a) works on large text corpora in the web domain employing corpus statistics to ensure tagging of entities in a text. It uses the TAP ontology (Dill et al., 2003b), and has been used to annotate about 264 million web pages with 550 million labels. The authors in (Zavitsanos et al., 2010) introduced a natural language processing approach where a semantic relatedness between the words in a document and the concept is calculated using exact, stem and semantic matching. In particular, they used WordNet synset path distance to measure the similarity of the text and the concept. The problem with this approach is that concepts from an ontology like Eurovoc may not explicitly appear in a text. In such a situation, a word-word similarity approach proposed by the authors would fail grossly. Charlton et. al. (Charton and Gagnon, 2012) on the other hand introduced a Wikipedia-based disambiguation technique for semantic annotation. They used Wikipedia pages to build metadata for each concept. The metadata consist of (1) surface forms, i.e., the links on a Wikipedia page and (2) the tf-idf weighted terms that a page is composed of. They also associate these concepts with DBPedia concept in order to provide a kind of semantic enrichment.

Even though we also incorporate explicit concept expansion to aid semantic understanding of the concepts, our work is significantly different owing to the manner of the use of Wikipedia and other resources to build a semantic profile for the concepts. In this regard, our work follows the approach of Gabrilovich (Gabrilovich and Markovitch, 2007; Egozi, Markovitch, and Gabrilovich, 2011) in the way we make use of the Wikipedia concepts to distill the concepts from Eurovoc. Our approach is however different in how we compute the semantic representation of the text segments and concepts.

Perhaps an abuse of term, it is important to describe what we defined here as semantic annotation in order to distinguish our work to the existing work. The reviewed works have focused on entity annotation and identification of mentioned subjects in a text which share semantic relationship with a list of concepts in a knowledge base. Specifically, the focus has been on Information Extraction (IE) rather than IR. Here, we focus on developing ways by which we can approximate and represent the meaning of an abstract concept and finding a correspondence between this representation and that of any selected text. In a way, SA as defined in this work is a task of semantic matching between two pieces of text, rather than labeling entities in a text with some semantic concept.

Generally, this kind of semantic matching can aid a structured organization of documents for an optimized search. For instance, users may search information by well-defined general concepts that describe the domain of information need rather than use keywords.

3.6 Conclusion

In this chapter, we describe our approach to segmenting text into topical sections in our attempt to motivate a fine-grained retrieval which reduces the problem of information overload. We reviewed the state of the art systems and also describe our definition of semantic annotation task.

Chapter 4

The E-Discovery Information Retrieval

In this chapter, we discuss the E-Discovery process and the predictive coding with specific focus on the IR part of the general E-Discovery model, i.e., review for relevance and privilege. Next, we motivate the reasons for our Neural Network Relevance model, a classifier for the E-Discovery task.

4.1 E-Discovery

Imagine looking for a thousand relevant documents out of a million candidate documents that are likely to include the thousand documents being sought. Rather than being a fiction, the above vignette captures what information seeking looks like in the era of big data, and how lawyers are expected to swim endlessly in the ocean of electronically stored information (ESI). Specifically, searching for evidence in an unstructured information is cumbersome, particularly when what is being searched is not exactly known. Just eight years ago, it was estimated that there are 988 exabytes of data in existence, and as Casey (Auttonberry, 2013) puts it, it would stretch forth and back from the Sun to Pluto if put in the paper form. This is already 2017, and I reckon that this sheer amount would have doubled, if not quadrupled. Conversely, organizations now process and store information more than ever.

Even though this explosion of data is particularly not a problem for the Legal domain or the Legal experts, the *Law*, as we know it postures as a means of conflict resolution. Just as in any human-managed society, organizations do have conflicts between one another, the end-product of which are usually litigations where the Law and its ordinances are brought to bear in providing amicable solutions. Parties involved in a litigation naturally would look for ways to strengthen their case, this usually involves the *civil discovery* process, in which a party requests the opposing party to produce documents that are in that party's possession, custody, and control which are pertinent to a case (Oard and Webber, 2013). As the name suggests, criminal litigation is not subject to the discovery process.

4.1.1 Federal Rules of Civil Procedures

In the United States of America (US), litigants are empowered to lodge *requests for production* based on the Federal Rules of Civil Procedures (FRCP)[1]. As we have explained in chapter 1, when a discovery process entirely involves ESI, it is called *E-Discovery* (Oard and Webber, 2013). E-Discovery in particular arises from the 2006 amendments to the FRCP. The *Rule 34* of the FRCP (*2006*) is reproduced below:

(***a***) *In General. A party may serve on any other party a request within the scope of Rule 26(b):*

1. *to produce and permit the requesting party or its representative to inspect, copy, test, or sample the following items in the responding party's possession, custody, or control*

   1.1. *any designated documents or electronically stored information—including writings, drawings, graphs, charts, photographs, sound recordings, images, and other data or data compilations -stored in any medium from which information can be obtained either directly or, if necessary, after translation by the responding party into a reasonably usable form; or*

   1.2. *any designated tangible things; or*

2. *to permit entry onto designated land or other property possessed or controlled by the responding party, so that the requesting party may inspect, measure, survey, photograph, test, or sample the property or any designated object or operation on it.*

(***b***) *Discovery Scope and Limits.*

1. *Scope in General. Unless otherwise limited by court order, the scope of discovery is as follows: Parties may obtain discovery regarding any non-privileged matter that is relevant to any party's claim or defense—including the existence, description, nature, custody, condition, and location of any documents or other tangible things and the identity and location of persons who know of any discoverable matter. For good cause, the court may order discovery of any matter relevant to the subject matter involved in the action. Relevant information need not be admissible at the trial if the discovery appears reasonably calculated to lead to the discovery of admissible evidence. All discovery is subject to the limitations imposed by Rule 26(b)(2)(C)*

What this means is that any non-privilege documents that are responsive to a production request[2] must be made available to the requesting party. A privilege document is a sensitive document whose disclosure could either expose the strategy adopted by a legal counsel and the client(e.g., client-attorney communications) or prejudice the producing party's interests (Oard and Webber, 2013). Common examples are the attorney-client

[1] https://www.federalrulesofcivilprocedure.org/

[2] Throughout the thesis, we have interchangeably used the words "Request for production", "Production request", "RFP", and "Request". Unless otherwise stated as having a different meaning, we have used them in reference to the same thing.

privilege or an attorney work product, according to *Federal Rule of Evidence 502*. By implication, lawyers involved in a lawsuit are exposed to hundreds of millions of documents which are to be reviewed for *privilege* and *relevance* often by the plaintiff and the defendant, with the attendant exorbitant cost. However, there are some exceptions where disclosure does not hold. For instance, FRCP (*Rule 26(b)(2)(C)(iii)*) requires that in the event that the requested information is not *'reasonably'* accessible, a court can limit discovery, or the parties are relieved from disclosure (*FRCP Rule 26(b) (2)(B)*). A tenable reason is if the disclosure would incur a high cost, is stored in an obsolete media, or creates an undue burden on the defendants. Another credible reason is the case of *proportionality rule* (*Rule 26(g)(1)*), where the cost of a proposed discovery exceedingly outweighs the potential benefit considering the needs of the case, the amount in controversy, the parties' resources, the importance of the issues at stake in the action, and the importance of the discovery in resolving the issues (Grossman and Cormack, 2010). Notwithstanding, FRCP *Rule 37(a)(4)* strictly penalizes partial or incomplete disclosure, for it states that "an evasive or incomplete disclosure, answer or response must be treated as a failure to disclose, answer, or respond[,]". These rules show how the Law frowns against *evidence hiding* while also protecting against harassment or intimidation of an opposing party by, for example, a big corporation (*defendant*) who could overwhelm the opposing party with an unprecedented amount of data in order to force an out-of-court settlement.

Perhaps, it is important to understand that the visibility or accessibility of requested data during E-Discovery is beyond any territorial or geographical boundary. Other than the exceptional cases where disclosure may not hold as highlighted above, the FRCP rule demands total compliance irrespective of the location in the world where the data may be domiciled. This also alludes to the heterogeneity of the kind of data involved in an E-Discovery process. Moreover, apart from the fact that this specific obligation often comes in conflict with the *privacy* law of some countries where discovery is alien, e.g., in Common law countries such as Canada, Australia, and United Kingdom (UK), it also conflicts the European Union (EU) Data Protection Directive *94/46/EC* (EDD), which was crafted to protect the privacy and protection of all personal data collected for or about citizens of the EU, especially as it relates to processing, using, or exchanging such data. The EDD encompasses all key elements from *article 8* of the European Convention on Human Rights, which states its intention to respect the *rights of privacy* in personal and family life, as well as in the home and in personal correspondence (Monique, 2011). Technically, this is also an important issue to be wary of during civil discovery as there could be *territorial/jurisdiction* conflict.

4.1.2 The E-Discovery Model

After receiving a request for production (RFP), a defendant is usually expected to inaugurate an E-Discovery information technology (IT) team whose task is to interface between the corporate counsel and the prosecuting counsel. The team in charge of the data is then interviewed to ascertain the available and relevant data. Next, a “Litigation Hold” letter is sent to all relevant parties in order to foreclose any alteration or destruction of the ascertained relevant data. A *Litigation Hold* or a *Legal Hold* is a communication issued as a result of current or reasonably anticipated litigation, audit, government investigation, or other such matter that suspends the normal disposition or processing of records. After this process comes the *'meet and confer'* meeting with opposing counsel and the Court. Here, the scope of production of ESI and the activity duration is negotiated. Furthermore, the negotiation often includes thorough agreement on the search techniques, keywords to be used in case of keyword search, the format for the production of ESI (TIFF or Native or both), the requirement for preservation of metadata, the *clawback agreements* (a clawback agreement is an agreement outlining procedures to be followed to protect against waiver of privilege or work product protection due to inadvertent production of documents or data); and issues of cost shifting (*Rule 26 (f)*) (Monique, 2011). This set of stages which highlight the information processing procedural activities has been codified into what is called the E-Discovery Reference Model (EDRM). Figure 4.1 shows the stages involved in an E-Discovery process. The authors (Oard and Webber, 2013) already elucidate these stages in their review work on E-Discovery. We highlight the important ones below:

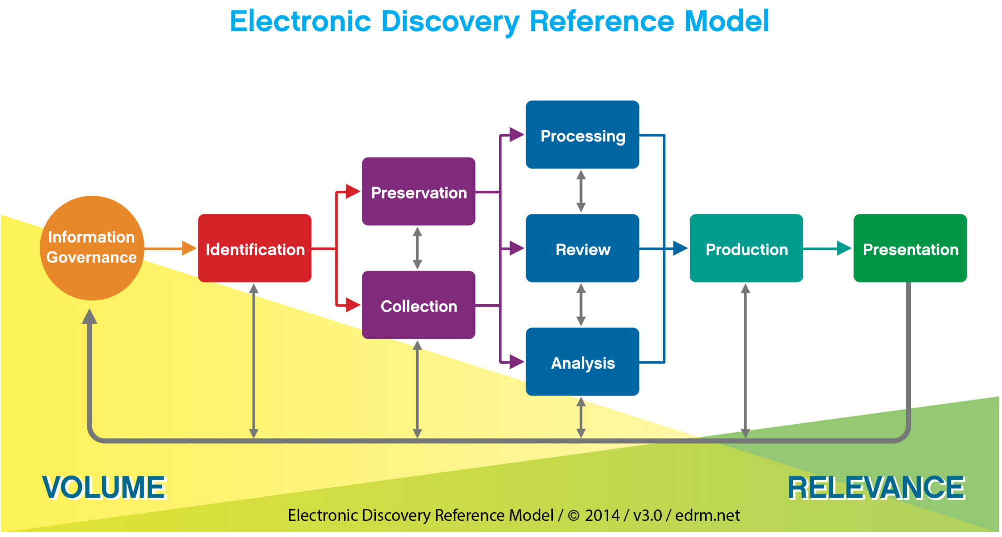


FIGURE 4.1: The E-Discovery Reference Model

- Information Governance: This deals with the coordination of all information processing activities before the litigation process. Common tasks include records management (e.g., to meet legal, regulatory or policy goals), archival storage of records that are appraised as having permanent value, information processed using personally owned devices such as smartphones or home computers, and information managed by other providers (e.g., the intent is to encompass all of the regular information processing activities of an organization prior to the start of an e-discovery process(e.g., “cloud services”).

- Identification: This involves locating the ESI that is relevant to the litigation without breaching the privileges of the organization. Usually, lawyers and E-Discovery information technology (IT) team constituted by an organization are usually involved. Two main tasks are involved, i.e., *data mapping* and negotiating the *discovery scope*. In the former, a data map showing the information flow in an organization is produced, while in the latter, parties agree about what information is to be collected from each data source and the restrictions to be followed. This process has been likened to the federated search in the general IR parlance.
- Collection and Preservation: Collection entails using particular techniques to gather the *identified* information, e.g., querying a database for information or using forensic techniques to recover an otherwise corrupted or inaccessible information. As Oard et. al. (Oard and Webber, 2013) puts it, preservation entails "maintaining the bit-stream, maintaining the information necessary to interpret the bit stream, and maintaining evidence of authenticity for the bit-stream".
- Processing, Review, and Analysis: This is where most IR work takes place. During *processing*, some operations are performed on the collection in order to format it into a desirable form for use by either the manual reviewer or a Technology Assisted Reviewer (TAR). During review, an expert (e.g., a lawyer) assesses each document one after the other for relevance if manual reviewing is done. The person may use Boolean search or keywords to initially weed out irrelevant documents. On the other hand, predictive coding may also be used, where reviewers manually inspect a sampled set of documents for relevance and then use the identified relevant documents as a *seed set* for a machine learning classifier.
- Production: This stage involves the delivery of documents that are deemed responsive from the review process. The produced documents must contain no privileged information. Usually, the produced documents are handed over to the requesting parties, accompanied with information about documents that have been with-held due for containing privileged information.
- Presentation: Here, further information could be deduced from the produced documents for further legal analysis as may be required.

4.1.3 Information Retrieval-Centric Electronic Discovery Model

The EDRM have a broader view of all the activities involved in the discovery process. A few of these activities are not of direct interest to an IR researcher whose focus is solely on the information retrieval aspect, as it is the case in this thesis. Oard et. al. (Oard and Webber, 2013) further presents an IR-Cetric model as shown in figure 4.2. This shows a waterfall view of the IR activities involved in the discovery process, leading to a different retrieval result at each stage, beginning from the *formulation* stage to the *sense-making* stage.

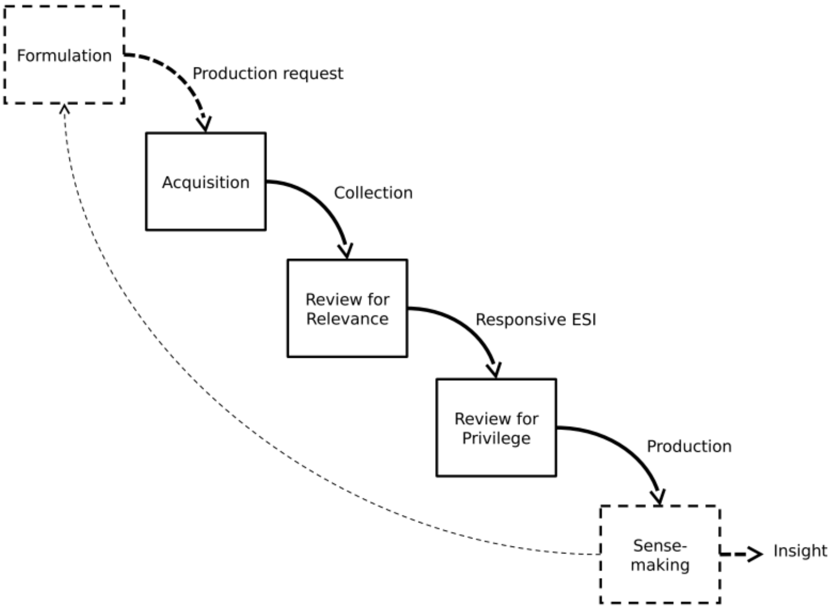


FIGURE 4.2: An IR-Centric View of The E-Discovery Reference Model. Dashed lines indicate requesting party tasks and products, solid lines indicate producing party tasks and products.

Three of these stages are particularly of interest in this thesis. The first is the Formulation stage where the request for production is received in the form of topics. Unlike in ordinary IR, topics do not essentially come in form of query. Rather, the producing party has to analyze and interpret the topic in order to determine what the query terms should look like. The request is then reformulated into a more stable query with the aid of query expansion and similar techniques (Xu and Croft, 1996; Voorhees, 1994). In our work, we utilize a form of explicit semantic analysis as described in chapter 5 in order to expand the reformulated topics. The other two important stages are the *review for relevance* and the *review for the privilege*. The two exhibits a core IR task, i.e., given some documents, determine whether any of the document is responsive to a given query or a given privilege search term. The task here can be likened to a text classification task with two classes, i.e., *Responsive* or *Non-Responsive*. This is also where E-Discovery seems to differ to most IR procedures where the goal is to produced a ranked list of relevant documents to a query (Salton, 1971).

In E-Discovery, seemingly determining whether a document belongs to either of these two classes suffices. However, it is also possible to rank the documents classified to be relevant based on their *relevance probabilities*. In this case, the goal is to push the most relevant document to the top such that documents with the highest relevance probabilities are presented for *production*. The legal track of the *Text Retrieval Conference* (TREC) organized by the *National Institute of Standards and Technology* (NIST) offers a ranked assessment for both the Interactive and Batch tasks (Cormack et al., 2010). Once the responsive documents have been identified, they can also be reviewed for privilege. This also can be likened to a binary classification, i.e., whether a document belongs to the class *Privilege* or *Not Privilege*.

Approaches for conducting review are the manual review, linear review, keyword search, or the TAR which is the focus of this thesis (Baron et al., 2007; Oard and Webber, 2013). Irrespective of the method that is adopted, considering the heterogeneous nature of the ESI data, it is necessary to perform an initial de-duplication routine. *De-duplication* helps to identify a canonical version of each item, such that the location of each distinct record is recorded against the item. This serves to reduce redundancy by removing duplicated items such that the same item would not be reviewed many times. Technically, it reduces the collection size and saves time, effort and costs attributed to repetitious reviewing (Oard and Webber, 2013).

The final part is the *sense-making* which as Oard (Oard and Webber, 2013) describes entail asking the *'5 W'* questions, i.e., *Who* were involved and what were their roles; *What* happened and what objects were involved; *When* did an event happen, including the sequencing; *Where* is an item located; and *Why* combines knowledge from the previous questions to provide a veritable and all-encompassing answer. Typically, the cost of conducting a manual review takes a significant portion of an entire discovery process (Grossman and Cormack, 2010).

While the general IR task is loose regarding the unit of retrieval, E-Discovery operates at the *document family* level, where a family constitutes a logical single communication of information, even though, the information could spread over many individual documents. For example, it makes sense not only to consider an email alone but also its attachments. Furthermore, an email with multiple replies may be grouped into a thread for they are likely to contain a continuous line of communication or topic. In particular, email, forum, and collaboration platforms constitute roughly *50%* of ESI in existing E-Discovery procedures (Baron et al., 2007).

4.1.4 E-Discovery Vs Traditional IR

E-Discovery has some distinguishing characteristics when compared to the general IR process. We itemize some of these features elucidated by (Oard and Webber, 2013):

- E-Discovery places emphasis on fixed result sets instead of ranked retrieval, e.g., the decision is whether a document is responsive or not responsive (classification *v.* Ranked).
- E-Discovery is recall-oriented rather than fixated on high precision as in web search.
- E-Discovery measures both absolute effectiveness of the retrieval system, as much as relative effectiveness.
- E-Discovery provides a nexus between the IR field and other exploratory fields like computer forensics and document management.

- The result from an E-Discovery process is greatly impacted by the level of cooperation between the plaintiffs and the defendants.
- The request for production is not an explicit query as in general IR.

4.2 The Case for Predictive Coding in E-Discovery

Generally, the review process may take several forms depending on the choice of the organization involved. The traditional approach is to employ lawyers for manual review. However, apart from the fact that the review process may take a long time, it is usually an expensive process. For instance, at an average rate of *50* documents per hour at $*50*, it would take a team of *50* lawyers about *400* days to review *10 million* documents assuming they work *10* hours per day, requiring a budget of $*10 million*. Even if the number of reviewers is doubled, conducting reviews for over *7* months would definitely take its toll on the litigation. Typically, the price for estimating this example is conservative, the Sedona Conference Commentary affirms that the billable rates for junior associates at law firms now starts at over $*200* per hour as at the year *2007* (Baron et al., 2007); a collection will most likely be bigger in real life, e.g., the *United States* v. *Philip Morris*, in which government lawyers had to search a database of *32* million Clinton-era White House e-mail records; and according to prior studies, an expert will most likely review just under 25 documents per hour (Roitblat, Kershaw, and Oot, 2010). Moreover, the cost of a litigation is not limited to just the discovery, for E-Discovery typically take just around *25*% of the actual litigation cost.

Owing to some of these constraints, attorneys have used a couple of search techniques to reduce the search space. Prominent among the techniques are the Keyword search, Boolean search, and Concept search. Boolean search, in particular, was popular with lawyers for it gives them the power to formulate queries with logical operators. Also, the fact that they are domain experts help in how they formulate queries and simplify search with the use of proximity operators. However, as we have explained in chapter 2, Boolean search mostly defaults to keyword search and it suffers from language variability problems such as polysemy and synonymy. More importantly, researchers have discussed how attorneys typically miss out on many relevant documents when using this approach. An approach like the Boolean search is behind the *Lexis-Nexis* and *Westlaw* search systems which have support for full-text search and also ranked retrieval for lawyers. For instance, Blair and Maron (Blair and Maron, 1985) show that attorneys using search tools based on Boolean search could only retrieve *20*% of the responsive documents even though they were convinced that they retrieved over *75*% of the responsive document. This illusion could be catastrophic considering that in E-Discovery, the cost of missing out on a single responsive document far outweighs that of producing non-responsive documents. Moreover, empirical studies have shown that Boolean search is strongly inefficient in large scale full-text search (Sormunen, 2001).

The success of *Machine Learning algorithms* (MLA) in text classification buoyed the interest of E-Discovery research community in exploring text classifiers for the discovery process. A linear classifier like the Naive Bayes algorithm was first explored. Later, Support Vector Machine (SVM) (Cortes and Vapnik, 1995) proved particularly adept for classification tasks. As explained in chapter 2, there are two basic paradigms in machine learning, i.e., the *supervised* which can be described as *learning-by-example* because the algorithm observes pattern from an example set (train) which is deemed to be the gold standard, and the *unsupervised* approach where the algorithm automatically infers pattern from the data without needing any example, e.g., clustering algorithm. The classifiers used are mostly supervised because they give better approximation. Because of this, Lawyers would, via *sampling*, select some documents which are responsive to a request for production (RFP), and these documents are coded as the *seed set*[3] for the MLAs. A sampling can be done through basic keyword search or Boolean search for the subset of data, and then review by the human expert. The seed set would also contains some non-responsive documents to a RFP. The MLAs then learn separate patterns about the responsive and non-responsive documents such that when a previously unseen document is introduced to the MLA, it is able to draw a margin that separates the document into either the responsive or non-responsive class. SVM in particular operates in this manner (Cortes and Vapnik, 1995). The use of MLAs in E-Discovery process has been termed *Predictive Coding* and put in another way, Technology Assisted Review. In general, Predictive coding techniques are iterative in nature, often going through a continuous process of refinement and correction[4] until the algorithm shows to satisfy a minimum expected accuracy. Once the expected accuracy is reached, the system is then deployed to perform classification on the test set.

As Roitblat (Roitblat, Kershaw, and Oot, 2010) shows, MLAs performed much better than human experts in classifying responsive documents. The results from the TREC legal track have also confirmed the assertion that TAR performs better than exhaustive manual review by human experts (Grossman and Cormack, 2010; Cormack and Grossman, 2014). Even though many legal experts were initially apprehensive of this technology, some have designated it as a *destructive* technology while a few already cast aspersion as to its reliability (Remus, 2013), however, studies have shown that predictive coding is able to drastically reduce E-Discovery costs by up to *71%* while maintaining search quality (Auttonberry, 2013). Perhaps, this efficacy has led to its recognition and legitimization by the court for use in litigation as pronounced herein - "predictive coding now can be considered judicially-approved for use in appropriate cases,"[5].

[3] The seed set could be likened to the relevance judgement for the train set in general machine learning task.

[4] In machine learning, we say tuning with the development set. This tuning could be in form of parameter optimization or model fine-tunning etc.

[5] See, e.g., Moore v. Publicis Groupe, 287 F.R.D. 182 (S.D.N.Y. 2012); In re Actos (Pioglitazone) Prods. Liab. Litig., No. 6:11-md-2299, 2012 WL 6061973 (W.D. La. July 27, 2012)

4.2.1 Other Applications of Predictive Coding in Litigation

Apart from being used as binary classifier for identifying whether a document is responsive or not, MLAs may also be employed to support other litigation aspects. Hampton (Hampton, 2014) identifies a few ways by why MLAs could be used nominatively. He opined that attorneys may used predictive coding to:

1. Identify key strengths and weaknesses in a client's case during early case assessments and preliminary investigations.
2. Streamline aspects of document review when responding to document requests.
3. Analyze a document production received from an opposing party or a third party.
4. Prepare for depositions, expert discovery, summary judgment motions and trial.

4.2.2 Advantages of Predictive Coding

The author (Hampton, 2014) also elucidates on the merits and demerits of predictive coding, some of which are highlighted below. Predictive Coding can:

1. Drastically reduce the number of documents requiring attorney review, thus saving time and cost, and in general improve the effectiveness of the process.
2. Minimize or eliminate the inconsistent production and privilege calls that plague every large document review and allow for a higher level of consistency in the process.
3. Identify more relevant documents than the traditional linear attorney review in which documents are reviewed one after another.
4. Substantially reduce the risk of being accused of deliberately hiding relevant documents, since it is far easier to justify the non-production of an important document where the predictive coding program coded it as non-responsive.

4.2.3 Disadvantages of Predictive Coding

Predictive Coding technique is not a one-stop-gap, it has some demerits which we highlight below:

1. Many coding protocols (including the one implemented in this thesis) operates on text without being able to analyze other file types, e.g., Spreadsheet, videos, etc. In E-Discovery, evidence could be hidden in this kind of files other than mere text.
2. In the case where an opposing counsel insists on joining the defendant team for seed set document coding, the opposing counsel may inappropriately gain access to a privilege information.

3. The success of MLAs depends on the quality or the validity of the seed set. An erroneous seed set or training process will cascade those errors throughout a production. Therefore, the process of coding a *seed set* requires the expertise of experienced attorneys. Specifically, there are existing studies which show how variance in relevance judgment may affect the performance of MLAs in an E-Discovery process (Voorhees, 2000; Wang and Soergel, 2010; Webber and Pickens, 2013; Grossman and Cormack, 2010).

4.3 The TREC-Legal Track

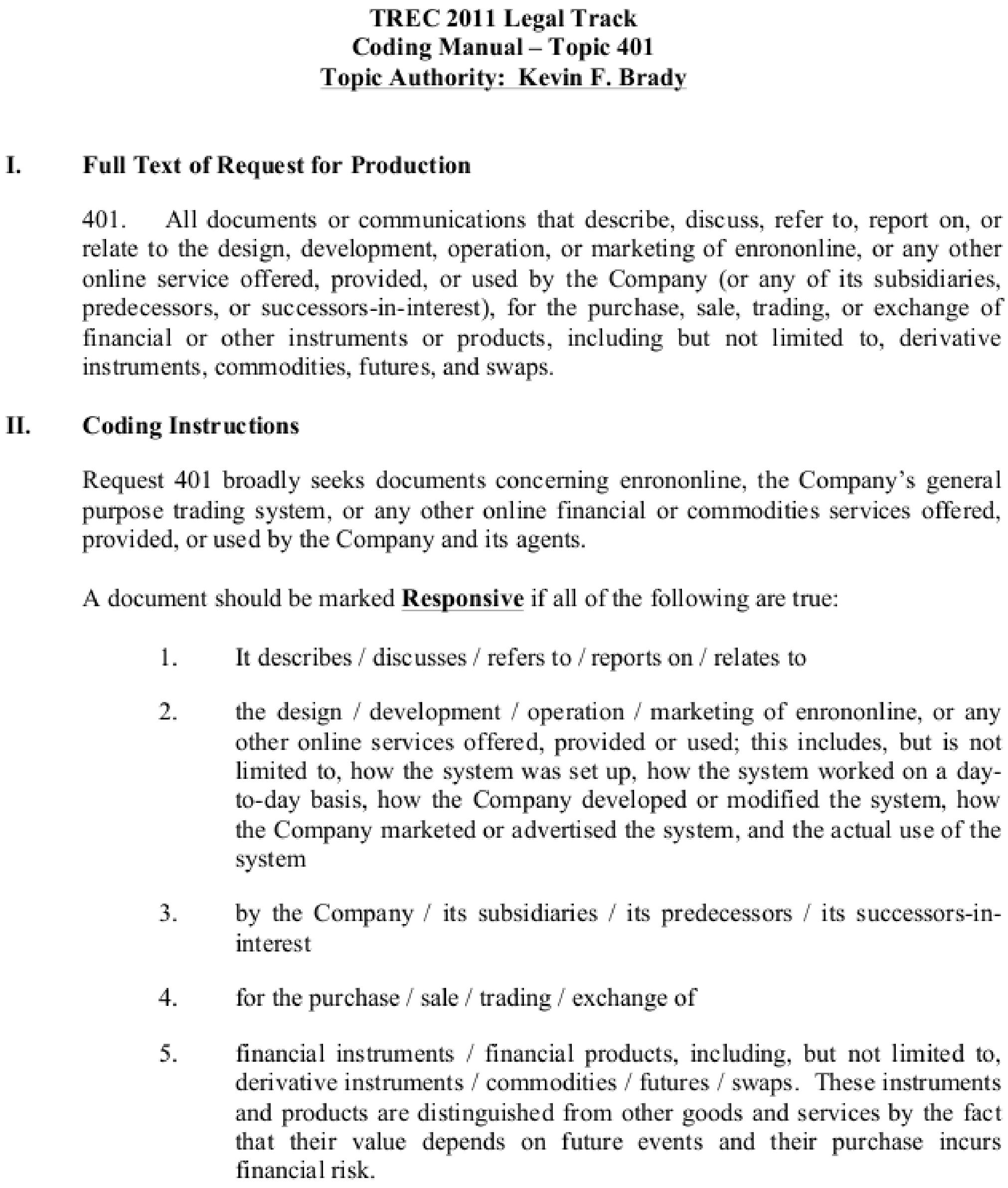
TREC is an annual event organized by NIST. The broad goal of TREC is to motivate large-scale IR among researchers and also providing a nexus between the academia and the industry where the techniques may find real-world application. TREC has provided large test collection and appropriate evaluation techniques to encourage research in this line. The legal track of annual TREC competition held for the first time in 2006 with the goal of creating an avenue where legal practitioners could interface IR researchers in providing an efficient search tool for large-scale legal search. The legal track mainly tries to simulate a real-world E-Discovery process, such that a large document collection is presented to participants who are to identify all the documents in the collection that are *responsive* to a RFP while reducing to the barest minimum the number of the *non-responsive* documents that are included in the responsive list for production. TREC Legal track has evolved over the years, however, the task can be divided into two either of which participants can elect to partake in. I highlight the two tasks below:

- **Learning task**: here, a set of seed set (coded as relevant and not relevant) is produced by the organizers. The seed set is then used either by a humans team or a MLA to estimate the probability of either relevant or not-relevant of other documents in the collection.

- **Interactive task**: here, both humans and technology are deployed in consultation with a *Topic Authority*, to classify documents in the collection as either Responsive or Non-Responsive, while also minimizing the number of false positives. In TREC 2010, this task builds on the *Batch* task of TREC 2009. This task also includes privilege review, i.e., identifying whether a relevant document contains sensitive information and should, therefore, be withheld from disclosure.

4.3.1 Request For Production

A Request for production is presented as a topic which directly relates to a complaint. It is possible that several RFPs are made regarding a single complaint. Figure 4.3 shows a RFP topic from the TREC 2011 Legal track. As we can see, the coding instruction gives the context that guides how a document may be identified as either responsive or not. Unlike



TREC 2011 Legal Track
Coding Manual – Topic 401
Topic Authority: Kevin F. Brady

I. Full Text of Request for Production

401. All documents or communications that describe, discuss, refer to, report on, or relate to the design, development, operation, or marketing of enrononline, or any other online service offered, provided, or used by the Company (or any of its subsidiaries, predecessors, or successors-in-interest), for the purchase, sale, trading, or exchange of financial or other instruments or products, including but not limited to, derivative instruments, commodities, futures, and swaps.

II. Coding Instructions

Request 401 broadly seeks documents concerning enrononline, the Company's general purpose trading system, or any other online financial or commodities services offered, provided, or used by the Company and its agents.

A document should be marked **Responsive** if all of the following are true:

1. It describes / discusses / refers to / reports on / relates to
2. the design / development / operation / marketing of enrononline, or any other online services offered, provided or used; this includes, but is not limited to, how the system was set up, how the system worked on a day-to-day basis, how the Company developed or modified the system, how the Company marketed or advertised the system, and the actual use of the system
3. by the Company / its subsidiaries / its predecessors / its successors-in-interest
4. for the purchase / sale / trading / exchange of
5. financial instruments / financial products, including, but not limited to, derivative instruments / commodities / futures / swaps. These instruments and products are distinguished from other goods and services by the fact that their value depends on future events and their purchase incurs financial risk.

FIGURE 4.3: Requests for Production (RFP) given as a Topic for the TREC legal track E-Discovery

the general IR, a topic is not an explicit query, and it is necessary to carefully digest its information in order to arrive at a set of valid query terms. The complaint information is a detailed description of a court filing which gives necessary background information that may help in coding a topic or enriching the topic during query reformulation.

4.3.2 Document Collection

The TREC document collection was derived from the EDRM Enron Dataset version 2. The document has been prepared by ZL Technologies and the organizers of the track Legal track. This collection contains around 1.3 million Enron email messages from Lockheed Martin (formerly Aspen Systems) who captured and maintain the dataset on behalf of FERC. The organizers make available the dataset in two formats, i.e., XML and PST (a Microsoft proprietary format employed by most commercial tools). Both versions contain a text rendering of each email message and attachment, as well as the original native format with a size roughly 100GB uncompressed. Both law students and professional lawyers were employed for reviewing the documents for the sampled set used for relevance judgment. According to the organizers, for the Learning task, 78,000 human assessments were used while for the Interactive task, 50,000 human assessments were used. The organizers already performed de-duplication on the dataset released for TREC 2010, yielding a total of *455,499* canonical documents and *230,143* attachments. This implies that the 1.3 million documents have been reduced to a total of 685,592 documents. Transitioning from scanned documents to email messages have also reduced random noise in the data which is usually introduced when converting a scanned document to text. See further description of the tasks and data in (Cormack et al., 2010). In general, each document is assigned an identifier (i.e., *doc-id*), likewise, each topic/query is associated with an identifier (i.e., *qid*). A relevance judgment is also provided. The relevance judgment contains multiple pairs of *qid* and *doc-id* along with the associated binary relevance label showing whether the document with a certain *doc-id* is relevant or not for the topic with a particular *qid*.

In the next section, we motivate the rationale for our relevance-matching Neural Network model for E-Discovery.

4.4 The Significance Of A Relevance-Matching Model

Generally, IR can be viewed as a kind of semantic matching between a document and the query. In practice, we want to retrieve a document that is semantically similar to the query. However, as earlier explained, E-Discovery is technically different, for the goal is to determine the relevance of a document to a query, i.e., the interest is in determining those documents that are responsive to a RFP. Traditional information retrieval focuses on document and query terms, here, relevance is a matter of overlap between these terms. As we have described in chapter 2, even though this approach seems to be simplistic and fails in terms of synonymy and polysemy, it is a generalization that offers a window of how complicated algorithms may be developed.

Researchers have already proposed a couple of NN architectures for information retrieval (Mitra and Craswell, 2017a). Majority of these systems can be classified based on how they build relevance signals. For instance, the authors (Guo et al., 2016) classified them into two, i.e., 1) the interaction-focused based e.g., Hierarchical Attention Matching (Adebayo, Di Caro, and Boella, 2017a), Match Pyramid (Pang et al., 2016), Arc-II (Hu et al., 2014), and C-DSSM (Shen et al., 2014); and 2) the representation-focused based systems (Severyn and Mochitti, 2015; Yu et al., 2014; Huang et al., 2013; Palangi et al., 2016; Hu et al., 2014; Shen et al., 2014). In the former, some local interactions are induced between the input texts and a neural network is used to learn the hierarchical interaction pattern for matching. In the representation-focused model, a neural network is used to obtain a semantic representation for each text separately, matching and other approximations between the two representations are then carried out. For example, Palangi et. al. (Palangi et al., 2016) employed LSTM-RNN to build sentence representation for both query and document, while Arc-I (Hu et al., 2014) uses CNN.

Some researchers have also employed embedding for IR. The most important ones are the latent semantic embedding (Gao, Toutanova, and Yih, 2011) and the continuous embedding (Clinchant and Perronnin, 2013; Mitra and Craswell, 2017b; Mitra et al., 2016; Mitra, Diaz, and Craswell, 2017; Ai et al., 2016) architectures.

Most of these models have also been applied to many *text-matching* NLP tasks like Natural Language Inference, Paraphrase Detection, etc. Guo et. al. (Guo et al., 2016) opined that while the text-matching tasks involve semantic matching, ad-hoc retrieval involves relevance matching because unlike in the text-matching, the query and document are not homogeneous, and while the query would be short, a document could be arbitrarily long. Furthermore, input texts in the semantic matching tasks are characterized by their linguistic and semantic structure, i.e., they retain all the grammatical structure of sentences, on the other hand, even if we argue that a document contains multiple sentences, the query does not usually have any grammatical or linguistic link between the query terms.

As we have earlier explained, the equivalent of the query in E-Discovery is the RFP. Even though RFP maintains the grammatical structure of a sentence, it cannot be used as a query for it usually contains many irrelevant information, and a query reformulation process would have to be done. If an E-Discovery system is modeled like an ordinary Ad-hoc IR system then it loses the distinction and peculiarity of the E-Discovery task. In summary, while the text-matching and similar Ad-Hoc IR systems emphasize on 1) compositionality of word to derive a sentential meaning of the inputs, 2) similarity matching pattern between the inputs, and 3) a global matching requirement between the sentential representation of the inputs; we observed that the E-Discovery task emphasizes on 1) an exact matching signal between the document and the query, i.e., word overlap and BOW features are still relevant in IR, 2) query term importance to avoid a *topic-drift* , 3) Semantic and Relatedness mapping between the document terms and the expanded RFP, and lastly 4) scope-based matching to compensate for the looseness in RFP and the bias for a longer document over the shorter ones. Therefore, a relevance-matching model must be flexible enough to be able to search for relevance signals within the local and global scope of the document. Most importantly, the model must in a unique way look for interesting parts of the document that show a semblance of semantic relatedness to the RFP.

The relevance model proposed in this thesis generates the semantic representation of document and query in a way that the focus is on semantic relatedness of the RFP representation across different points in a document without dismissing the term matching signals. To achieve this, we introduce many semantic and lexical features which are extracted by separate component neural networks in order to model relevance in a simplistic way. In summary, our model is an ensemble feature-rich approach that incorporates relevance score using a traditional BOW approach (TFIDF), a latent semantic model (LSA), a representation-focused model, an interaction-focused model, a continuous embedding distance model, and lastly, a position-aware model for scope-based matching.

Part II

Concept-Based Information Retrieval

Chapter 5

Concept-Based Information Retrieval

In this chapter, we describe our semantic annotation framework and how the framework is used to obtain the semantic representations for concepts and document segments, and a similarity technique-based mapping between similar representations. In other words, a document segment that has similar semantic representation with a concept is annotated with that concept. In this way, documents can be indexed based on their conceptual properties. The significance of our work is how we learn word-concept distribution in a totally unsupervised way. Furthermore, our approach utilizes both the lexical and the semantic features which are obtained in the process of concept expansion and semantic representation of a concept.

Our approach can be divided into three key parts which are:

- Concept expansion and representation
- Document representation
- Concept-document mapping

We describe each of these steps below.

5.1 Concept Expansion And Representation

Legal concepts in an ontology usually do not have any explicit definition. The only way to extract the meaning of a concept is by finding alternative ways of expanding the concept. Concepts themselves are abstract ideas that some words may be used to describe. As earlier explained, the Eurovoc thesaurus has hierarchically organized concepts. Each of the concepts is a node which is identified by a label or descriptor. A node must have a parent and may have siblings as well as children. A simple way of annotating a concept with a document segment is by performing lexical matching, i.e., to check the occurrence of a concept descriptor in a text. However, a descriptor may not appear explicitly in a text. Furthermore, a concept descriptor may be composed of more than one word, i.e., it could be a bi-gram or n-grams. Here, we construct a profile for each concept. A concept profile is like a signature which incorporates all the descriptive information about a concept. We employ three strategies for expanding and representing a concept. These include: lexical expansion (with WordNet and a word embedding model), concept representation with Wikipedia document, and concept representation with Eur-Lex document. The combination of the individual representation obtained from these strategies form the profile of a concept.

5.1.1 Lexical Expansion With WordNet And Word Embedding

The first step here is to perform what we called *Concept Expansion*. Similar to *Query Expansion* (Voorhees, 1994), the essence of performing Concept Expansion is to enrich the concept with words that are semantically similar to it. The first approach is to use *WordNet* to obtain the synonyms of a concept while the second approach is to obtain a top-k[1] related words to the concept from a word embedding model. The use of the word embedding model is important since some concept may not be derived from WordNet. Also, a word embedding model like the pretrained *GloVe* (Pennington, Socher, and Manning, 2014) which condenses the distributional representation of around 840 billion words into a low dimensional vector space, where related words lie very close in the vector space, are useful for obtaining semantically similar or related words to any concept. The fact that an algorithm like GloVe is trained on billions of terms makes it possible to capture information than any human-generated thesaurus like the WordNet could ever capture. In particular, Researchers have shown that it captures semantic similarity and relatedness. Semantic relatedness in particular is an essential ingredient in emerging in emerging IR systems.

First, given a descriptor, we check for its synset in the WordNet. If a concept descriptor is not a unigram, we also check its occurrence in the WordNet, e.g., the word *Jet-lag* has two joint terms (*jet* and *lag*). However, it is a term in the WordNet. On the other hand, the concept *public-health* does not appear in the WordNet. In this case, we break the n-gram into its constituent terms, e.g., *public* and *health*. We then search for the top synonyms of the individual word in the WordNet as described in (Adebayo, Di Caro, and Boella, 2017c). Second, the same procedure is repeated except that instead of using the WordNet, semantically related words to an input word is obtained using the GloVe model.

Third, given that the concepts in a thesaurus are organized in a hierarchical manner, it makes sense to use the knowledge about this hierarchical structure to also enrich a concept. The idea is that the siblings of a concept are also more or less semantically similar to the concept. Likewise, the children nodes provide a more specific but less general term than the parent node. Most importantly, using this information automatically disambiguates a concept. Here, for any given concept node in a thesaurus tree, we traverse the tree in order to locate its dependents, which includes parent, siblings, and children. First, if a node has no child, we select its leftmost and rightmost siblings. Next, we traverse the tree up and select its parent. In the second case, if a concept node has one or more

[1] in our experiment, the parameter k = 3, is the number of topmost the synonyms to be selected

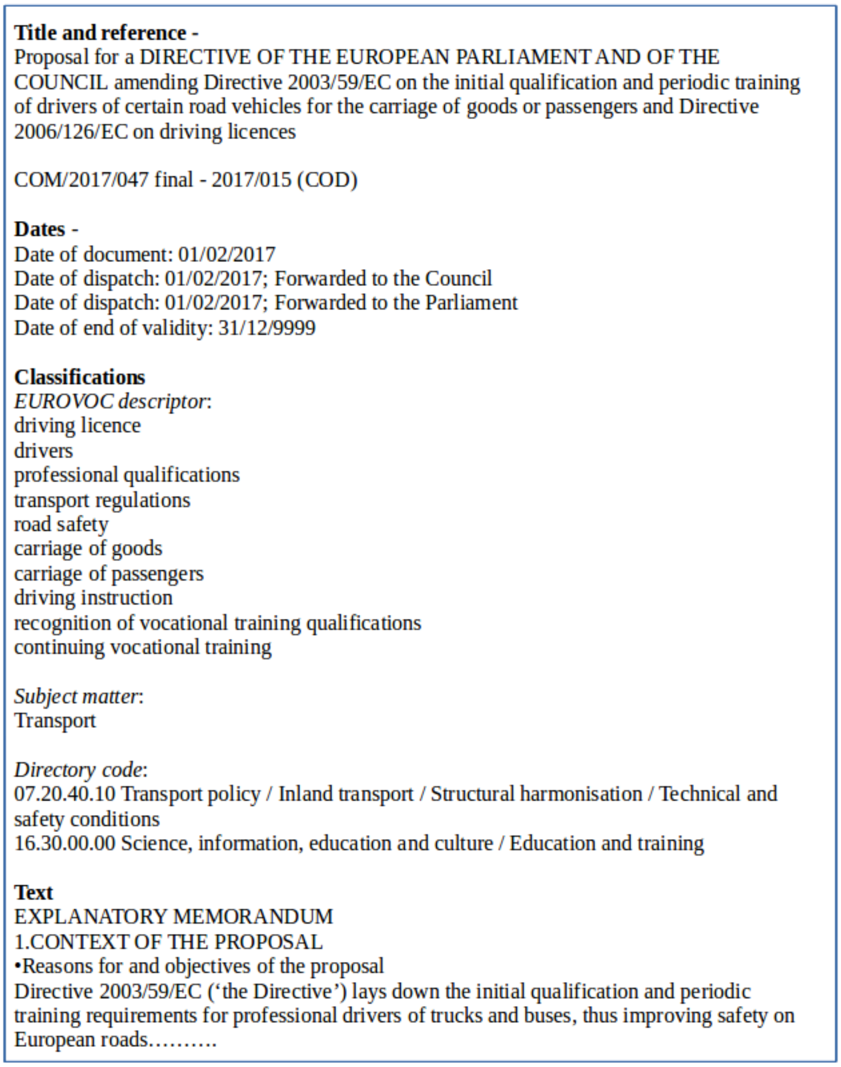
children, we only select all its children as well as the parent. Once these dependents for a concept are retrieved, following the steps described before, we obtain the semantically related word for each dependent concept using the GloVe embedding model and incorporate these synonyms into the profile of that concept. The set of related terms obtained from these lexical expansion approaches is called the *lexical-profile* of a concept.

5.1.2 Explicit Semantic Analysis Using EUR-Lex And Wikipedia Documents

The terms obtained in the lexical expansion phase may not fully capture the semantics of a concept. For instance, including many synonyms may introduce topic drift since those words may not have the same direct sense with the concept. A solution is to view a concept as a document, or more easily, a Bag-of-Words which contains the representative words for that concept. It is therefore important to identify some external knowledge resources which fully describe each concept. Deriving semantic information from an external knowledge base in this manner is referred to as Explicit Semantic Analysis (ESA), and researchers like Gabrilovich and Egozi (Gabrilovich and Markovitch, 2006; Gabrilovich and Markovitch, 2007) have utilized Wikipedia pages to obtain a semantic representation of an individual concept. In their work, each Wikipedia page[2] represents a concept, where the title of the Wikipedia page/document literary gives the concept being represented. In other words, the Wikipedia page/document title is a concept descriptor, and the assumption is that the words that are contained in that page gives a representation of this particular concept. The authors, therefore, considered a vocabulary consisting of all the words that appear on all Wikipedia pages. They then build two inverted indexes, i.e., one maps each word in the vocabulary to all the page titles (i.e., concept in this regard) of Wikipedia pages/articles where the word appears. The other inverted list maps each Wikipedia page title to all the words that are contained in that Wikipedia page/Article with that title. In a way, they have the conceptual representation for each word, as well as a term representation for each concept. They have successfully used this approach to measure the relatedness of concepts, and in particular, they adapt it for conceptual information retrieval (Egozi, Markovitch, and Gabrilovich, 2011). This approach is also similar to the exemplar-based concept representation approach described in (Noortwijk, Visser, and De Mulder, 2006), which has been used for conceptual legal text classification and ranking. In particular, Witten and Milne (Witten and Milne, 2008) utilized only the links on each Wikipedia page without considering the words contained in the page as representing the concept. By measuring the similarity of links in one page with those in another with a simple cosine-like formula, they are able to determine how related two concepts are. Since their algorithm uses only the links, they claimed that it is less computationally expensive. The approach described in (Hou, 2014) is also based on this technique.

[2] Here, Wikipedia Page, Document, or Article denote the same thing.

Even though Wikipedia concepts have been shown to be useful in some general NLP tasks, replicating this feat in the legal domain may be difficult. Legal experts conceive concepts in a normative way, whereas this specific attribute is lacking in ordinary documents, despite the fact that legal documents are also expressed in a natural language. Furthermore, legal documents have a particular formal nature and contain a lot of technical jargons for they are written in legislative terms (Mommers, 2010). If we strictly use Wikipedia concepts, then we lose some peculiarities of the legal jargons which the legal concept seeks to represent. In this work, we utilize EUR-Lex documents in combination with the Wikipedia documents to represent each concept. The EUR-Lex database consists of millions of articles. As we have explained in Chapter 1 and in the sections above, these articles are from different legal categories such as treaties, international agreements, legislation in force, legislation in preparation, case-law and parliamentary questions, etc, and are available in HTML or PDF format. Furthermore, each document in EUR-Lex is associated with some Eurovoc concept. On the average, a document has around 5.31



Title and reference -
Proposal for a DIRECTIVE OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL amending Directive 2003/59/EC on the initial qualification and periodic training of drivers of certain road vehicles for the carriage of goods or passengers and Directive 2006/126/EC on driving licences

COM/2017/047 final - 2017/015 (COD)

Dates -
Date of document: 01/02/2017
Date of dispatch: 01/02/2017; Forwarded to the Council
Date of dispatch: 01/02/2017; Forwarded to the Parliament
Date of end of validity: 31/12/9999

Classifications
EUROVOC descriptor:
driving licence
drivers
professional qualifications
transport regulations
road safety
carriage of goods
carriage of passengers
driving instruction
recognition of vocational training qualifications
continuing vocational training

Subject matter:
Transport

Directory code:
07.20.40.10 Transport policy / Inland transport / Structural harmonisation / Technical and safety conditions
16.30.00.00 Science, information, education and culture / Education and training

Text
EXPLANATORY MEMORANDUM
1.CONTEXT OF THE PROPOSAL
•Reasons for and objectives of the proposal
Directive 2003/59/EC ('the Directive') lays down the initial qualification and periodic training requirements for professional drivers of trucks and buses, thus improving safety on European roads……….

FIGURE 5.1: An excerpt of a sample EUR-Lex document with document ID:52017PC0047 showing the descriptors and a few metadata.

descriptors which it has been labeled with (Mencia and Furnkranz, 2010). Mencia and Furnkranz (Mencia and Furnkranz, 2010) selected some 19,348 documents from EUR-Lex which they used for their Eurovoc text classification experiment. The documents were selected from the English version of the Directory of Community legislation in force [3]. The EurLex dataset[4] consists of documents from the secondary law and international agreements. According to the author, the legal form of the included acts are mostly decisions (8,917 documents), regulations (5,706 documents), directives (1,898 documents) and agreements (1,597 documents). These documents have been labeled with 3,956 Eurovoc concept descriptors. Similarly, the Wikipedia is an example of a knowledge base with a vast amount of interlinked concepts, and it's freely available on the web. Because it is freely available and entirely contributed by volunteers from different fields and background through an open editing framework, the information provided is of substantial quality. As of 17 August 2017, Wikipedia contains 5,407,013 English articles. Although, the number of articles is around 42,726,999 when all the languages (293 in total) are considered. In total, about 27 billion words are contained in Wikipedia webpages which are managed by a group of 1251 administrators for the benefit of its more than 31 million users[5].

5.1.3 Modeling Concept With Eur-Lex Documents

Each document in the EUR-Lex dataset is labeled with some concepts, for example, Document *52017PC0047* has *10* concept labels which are {driving license, drivers professional qualifications, transport regulations, road safety, carriage of goods, carriage of passengers, driving instruction, recognition of vocational training qualifications, continuing vocational training }. Our method for building the explicit semantic representation of each concept from the Eur-Lex dataset is detailed below.

1. For each unique concept C_i in the Eurovoc thesaurus, construct a list of all the documents that it has been labeled with. We call this the concept bag, e.g. the concept bag for a concept is C_i = { D_{i1}, D_{i2},...., D_{ik} }. Where D_{ij} to D_{ik} is not an ordered sequence but rather, j,...,k represents an unordered unique ID of documents labeled with concept C_i. We also use this to build an inverted index or a dictionary Con_{dic} where the keys are the individual concept and the values for each key are the set of documents that are labeled with that concept. e.g., Con_{dic} = { C_1: D_1, D_3, D_5; C_2 : D_2, D_{15}, D_{23}; ....; C_n : D_a, D_b, D_c}.

2. Each D_{ij} is is a Bag-of-Word consisting of all the terms in the document. Each term in the document is from a vocabulary *V_eurl* consisting of all the unique words in the collection, i.e., all the words in each document of the EurLex dataset. For example, D_{ij} = { d_1, d_2, d_3, ..., $d_{|V_{eurl}|}$ }, where ∣V_eurl∣ is the number of words in

[3] http://eur-lex.europa.eu/en/legis/index.htm

[4] http://www.ke.tu-darmstadt.de/resources/eurlex/

[5] https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia

the vocabulary V, each word $\mathrm{d}_p \in V\_eurl$ is a unique term in the vocabulary, and C_i contains a sequence of word distribution from each D_{ij}.

3. We build a TF-IDF weighted sequence of each unique term d_p in the concept bag C_i. This is a distributed weight of each word d_p, capturing its overall importance to C_i based on its frequency in each $\mathrm{D}_{ij} \in \mathrm{C}_i$. Instead of a sequence of a set of words, we now have a single weighted sequence of all the unique terms in C_i, based on its importance to C_i.

4. Let the weight associated with each term d_p in the concept bag C_i be w_i such that $\mathrm{C}_i = \{ \mathrm{d}_1 (\mathrm{w}_1), \mathrm{d}_2 (\mathrm{w}_2), \mathrm{d}_3 (\mathrm{w}_3), ..., \mathrm{d}_p (\mathrm{w}_p) \}$. We rank each term in the descending order of their weight such that the most important words are ranked higher.

5. We do not want to use all the weighted terms C_i. In fact, this is the reason for the weight-based ranking. Because of this, we select the *top-s* weighted terms to use for describing the concept. The parameter *top-s* is a heuristically determined number which determines the number of the top ranking weighted terms to use in describing a concept. This parameter could be optimized by varying it based on the performance of the system. However, we have chosen a default value *top-s = 25*. The *top-s* ranked terms which are used to represent a concept are called the *Eurlex-profile* of the concept.

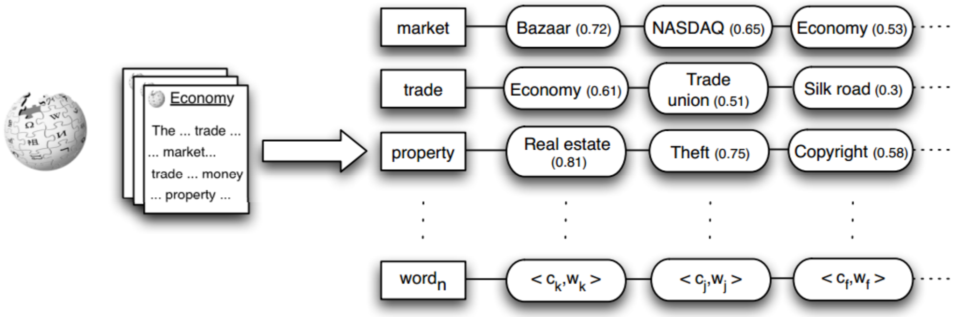


FIGURE 5.2: ESA generation from Wikipedia articles. The articles and words in them are processed to build a weighted inverted index, representing each word as a vector in the space of all Wikipedia concepts (articles) . (Source: (Egozi, Markovitch, and Gabrilovich, 2011).)

5.1.4 Modeling Concept With Wikipedia Documents

In the Wikipedia, unlike the Eur-Lex documents where a text is assigned several descriptor labels, each document is seen as a concept. Figure 5.2 shows how ESA is built from Wikipedia. Following the work of (Gabrilovich and Markovitch, 2006; Gabrilovich and Markovitch, 2007), our method for computing the semantic representation of a concept using Wikipedia is described below.

1. For each Eurovoc concept descriptor, we query the Wikipedia to check if a page exists for that descriptor. Note that the Wikipedia concept may not be written exactly

like the Eurovoc concept. For instance, the equivalent page for the Eurovoc concept 'Driving Licence' is named 'Driver's license', likewise, the Wikipedia concept for the Eurovoc concept 'Professional Qualifications' is 'Professional Certification'. However, this difference does not matter. Therefore, a concept bag C_i consist of a sequence of all the terms in the tokenized Wikipedia page.

2. It is possible that a Eurovoc concept does not have a page on Wikipedia. For example, there is no exact page on Wikipedia for the Eurovoc concept 'Driving Instruction' shown in the sample document in figure 5.1. In this particular example, 'Driving Instruction' or any missing concept is a node in the Eurovoc thesaurus and we traverse the thesaurus hierarchy in order to select the parent of that particular concept. We then use the Wikipedia page of the parent for this particular missing child node.

3. Similarly, we build a concept bag for each Eurovoc concept, this consist of the tokenized terms of the Wikipedia page for each concept. Also, because each concept in Wikipedia corresponds to a document, the concept bag here is not a sequence of sequence, i.e., a sequence of documents each containing a sequence of its tokenized terms. Rather, this contains just a sequence of the tokenized terms for each Wikipedia page per concept. All the terms contained in all the documents for all Eurovoc concepts form the vocabulary V_{wiki}.

4. Similar to step 3 of our concept representation with EUR-Lex documents, we build a TF-IDF weighted sequence of each term in each concept bag. The TF-IDF weight shows the importance of that term for that document.

5. We rank each term in the descending order of their weights and select the *top-s* ranked terms. Here, the default value for *top-s* = 25. The *top-s* ranked terms which are used to represent a concept are called the *Wiki-profile* of the concept.

Note that unlike in the work of (Gabrilovich and Markovitch, 2006; Gabrilovich and Markovitch, 2007; Hou, 2014) and (Witten and Milne, 2008), we did not make use of any extra meta-data or structure that is available in Wikipedia. For example, a Wikipedia page will normally contain an *info-box* which contains a summary of attributes of entities mentioned in a page. Furthermore, a Wikipedia page will contain many *in-coming* (links referencing this particular page/concept by another concept) and *out-going* links/*redirect pages* (links referencing other concepts directly from this concept), in addition to the classification *categories*, *disambiguation pages*, and *inter-language links*. However, since we only utilize ESA for concept expansion, exploiting this information will only introduce a lot of noise in the representative terms for a concept, and ultimately resulting in a semantic drift.

5.2 Obtaining the Overall Concept Representation

For each concept, we say that the combination of terms from the lexical expansion, the EUR-Lex document representation, and the Wikipedia concept representation forms the profile of each concept. The profile *profile*$_{con}$, which is the set of all descriptive terms of a concept is defined according to equation (5.1):

$$profile_{con}(a) = \{Lexical - Profile(a)\} + \{Wiki - Profile(a)\} + \{Eurlex - Profile(a)\} \tag{5.1}$$

The concept profile *profile*$_{con}$ for any particular concept may contain duplicated terms. This redundancy is useless and any repeating term in *profile*$_{con}$ is removed. Ideally, the goal is to obtain a semantic representation of a concept, such that this representation can be compared to any document or document unit. We obtain a semantic representation of a concept by representing each term in *profile*$_{con}$ with its corresponding vectors obtained from a word embedding model. We also utilized the GloVe embeddings here, such that each term in *profile*$_{con}$ is now represented by its 300-D vector. The overall semantic representation of a concept is obtained by vector averaging, i.e., summing all the vectors and normalizing by the total number of the word vectors according to equation (3.13). This yields a single vector $\overrightarrow{Sem - rep_{con}}(a)$.

5.3 Semantic Representation for Documents/Segments

Given a document that is to be annotated, we pass the document through our text segmentation module which divides the document into topical units. The goal is to obtain a semantic representation for each document unit. Researchers have shown that a fixed-length feature representation of a variable-length piece of text (e.g., sentences, paragraphs, section etc.) can be learned such that the fixed representation contextually captures the full semantic of the variable piece of text (Mikolov et al., 2013a). Paragraph vector relies on compositionality of vectors, e.g., vector averaging. The paragraph vector (Le and Mikolov, 2014) was proposed in this regard and Dai et. al. showed that it can be used to embed a full document (Dai, Olah, and Le, 2015). Instead of training a paragraph vector (Doc2Vec) model separately, we perform part-of-speech tagging for each segment using the Stanford POS tagger (Manning et al., 2014). For each segment, only the verbs, adjectives and nouns are retained for they carry more semantic information. We then perform vector averaging of the retained words in each segment according to equation (3.13), yielding a 300-D vector ($\overrightarrow{Sem - rep_{doc}}(seg_i)$.) which carries the meaning of each segment. Just like we have a semantic representation of each concept, we now have a semantic representation of each segment.

5.3.1 Concept And Document Mapping

Mapping concepts to documents or document segments can be viewed as the task of finding a semantic correspondence between the semantic representation of a concept to the semantic representation of each document segment. We formalize it as 4-tuple $(\overrightarrow{Sem-rep_{con}}(a)\ ,\ \overrightarrow{Sem-rep_{doc}}(seg_i^m)\ ,\ Rel\ ,\ COS)$, where $\overrightarrow{Sem-rep_{con}}(a)$ is the semantic representation of a concept *a*, $\overrightarrow{Sem-rep_{doc}}(seg_{i=0}^m)$ is the set of all the semantic representation of all document segments (*m* is the total number of document segments available), *Rel* is the semantic relationship between $\overrightarrow{Sem-rep_{con}}(a)$ and a particular segment $\overrightarrow{Sem-rep_{doc}}(seg_i)$, and it is computed with *COS*, which is the well-known Cosine similarity formula.

Matching a given concept to a text segment is, therefore, a simple semantic similarity task between the semantic vector of a concept and the semantic vectors of all document segments. In order to achieve this, we employed Faiss[6] for indexing. Indexing the vectors allows for easy similarity calculation. Once the similarity is calculated using the Cosine similarity formula given in equation (2.10), a ranking of the segments based on their similarity to a concept is performed and the concept is associated with all the segments with similarity above a particular threshold. The threshold is a parameter which is optimized by being changed randomly according to the annotation accuracy. As a default value, we recommend a value between 0.75 - 1.00, depending on the annotation task and the kind of documents involved.

5.4 Experiment

The system described in this chapter combines different standalone text analytics and processing components, which includes the text segmentation subsystem and the semantic annotation subsystem. We describe in detail the results obtained for each experiment.

5.4.1 Evaluating The Text Segmentation Module

Our text segmentation experiment uses the Choi's dataset, which perhaps, is the most frequently used dataset to evaluate text segmentation algorithms. Also, our baselines (Choi, 2000; Riedl and Biemann, 2012b; Hearst, 1997) have been evaluated on this dataset, which allows for an easy comparison. We used the P$_k$ error (Beeferman, Berger, and Lafferty, 1999) and WindDiff (Pevzner and Hearst, 2002) evaluation metrics which are commonly used. These two metrics measure the rate of error in segmentation with a lower value signifying better segmentation accuracy. Other common metrics are the IR based precision, recall and accuracy. However, these IR-based metrics over-penalize the *near-miss* scenarios, e.g., when an actual segment is wrongfully partitioned into two different segments

[6] Faiss is available at `https://github.com/facebookresearch/faiss`

Window	**3 - 5**	**6 - 8**	**9 - 11**	**3 - 11**
1	1.76	2.90	4.0	2.64
3	**0.89**	**1.18**	**0.49**	**0.67**
5	1.30	1.53	3.80	1.80

TABLE 5.1: Evaluation on Choi's Dataset using P_k error metric.

Window	**3 - 5**	**6 - 8**	**9 - 11**	**3 - 11**
1	1.82	2.94	4.21	2.68
3	**0.93**	**1.41**	**0.49**	**0.71**
5	1.29	1.48	3.87	1.82

TABLE 5.2: Evaluation on Choi's Dataset using WinDiff error metric.

by an algorithm. The LDA model utilized in our experiment was trained on the Brown corpus and a portion of Wikipedia dump[7]. We used the Gensim version of the LDA algorithm. Gensim is a python library for an array of NLP tasks [8]. Among other parameters, the number of topics specified for training is 50 and the training was concluded under 20 inference iterations.

We compare the result of our algorithm with the TopicTiling system (Riedl and Biemann, 2012b), a TextTiling based system which solely relies on topics assignment to document from LDA. We also compare the result with TextTiling and Choi's system as reported by Rield and Bielmann (Riedl and Biemann, 2012a). For all the reported results from other systems, we did not reproduce the experiments but instead, we reused the results reported in (Riedl and Biemann, 2012a).

Tables 5.1 and 5.2 shows the results of our algorithm on Choi's Text Segmentation dataset using the P_k and WinDiff error metrics, respectively. Each column shows the result obtained when the number of sentences is varied, e.g., 3-5 sentences, 6-8 sentences etc. We see that for both P_k and WinDiff metrics, our system obtained the best result when the window size = 3. Table 5.3 gives the comparison of our system against some state-of-the-art systems. Specifically, we selected TopicTiling (Riedl and Biemann, 2012a) algorithm

[7] The Wikipedia data was downloaded on July 30, 2015. It is accessible at https://dumps.wikimedia.org/enwiki/.

[8] It is available at https://radimrehurek.com/gensim/

Algorithm	**3 - 5**	**6 - 8**	**9 - 11**	**3 - 11**
TextTiling	44	43	48	46
Choi LSA	12	9	9	12
Topic Tiling	1.24	**0.76**	0.56	0.95
Our System	**0.89**	1.18	**0.49**	**0.67**

TABLE 5.3: Evaluation on Choi's Dataset showing comparison of our system to selected State-of-the-art Text-Segmentation algorithms.

Window	3 - 5	6 - 8	9 - 11	3 - 11
1	1.92	3.30	4.1	2.98
3	**1.19**	**2.23**	**0.82**	**0.91**
5	1.70	2.36	3.89	2.20

TABLE 5.4: Evaluation of our algorithm showing the impact of Boundary Adjustment on our system's performance. Evaluation was done on Choi's Dataset using the P_k error metric.

as it is the most similar to our work. The rationale for selecting the benchmark systems is well described here (Adebayo, Di Caro, and Boella, 2016e). Our intention is to show that our boundary-adjustment ideas really improves the performance of the system. The TextTiling and Choi's work have been severally outclassed by other systems (Du, Pate, and Johnson, 2015; Misra et al., 2009; Misra et al., 2011) but were selected based on their popularity. Moreover, TopicTiling also outperformed these systems. We see that our system clearly outperforms every other system within all sentence size variation, except at 6-8 where TopicTiling has a better score. To show the importance of the boundary adjustment component of our work, we reproduced our experiment without adjusting the boundary. Table 5.4 shows the effect of the boundary adjustment. Note the significant decrease in performance when boundary adjustment is not used.

5.4.2 Evaluating The Semantic Annotation Module

We selected 100 documents from EurLex website, 25 documents each from four different categories. EurLex is an open and regularly updated online database of over 3 million European Union documents covering EU treaties, regulations, legislative proposals, case-law, international agreements, EFTA documents etc. Documents are already classified using Eurovoc descriptors. We used the Eurovoc thesaurus as the ontology. The EurLex database as well as the Eurovoc thesaurus are both multilingual. Currently, it is available in 26 European languages. The documents downloaded are English versions from *Consolidated Acts* section of the website. Specifically, we selected documents under Transport Policy category. The sub-categories include {Transport Infrastructure, Inland Transport, Shipping, Air Transport}. The tiny size of the test data was informed by the level of human efforts required to perform human annotation. We evaluated the system on a task of *conceptual tagging*. Furthermore, we verified that these documents are not included in the original EUR-Lex dataset of (Mencia and Furnkranz, 2010) in order to avoid conflict of interest since we utilized the dataset for training.

Conceptual Tagging measures the performance of the system in correctly associating a text segment with a concept. We measured the performance of the system against annotations from human judgment. Many semantic-related tasks e.g., (Egozi, Markovitch, and Gabrilovich, 2011) have used human judgments in the past for humans have innate ability to ascertain how appropriate a text is to a concept. Human judgements can be used as the 'gold standard', which the result of an algorithm can be compared against. The assumption is that human judgments are correct and valid. To achieve this, all the documents were first automatically segmented into topical sections with our text segmentation algorithm. Two volunteer pre-annotators were then asked to read each segment and assign appropriate Eurovoc descriptors to the segments in the document. The descriptors chosen for each document are those which the document was labeled with on the EurLex website. Also, a segment of a document can only take a descriptor only from the one assigned to the document.

A segment may not be labeled with a concept descriptor if human annotators believed that there is no semantic relationship between it and any of the concept. Also, a segment can have more than one concept associated with it. A third volunteer compares annotations from the first two volunteers and where annotations do not correlate, decides the final annotations per document. It is observed that 13% of the annotations from the first two annotators were disputed and determined by the third annotator. The pre-annotators were volunteered Masters student of Law while the validator is a doctoral student of Law with a few years practice experience. The agreements were rated based on individual's judgment in labeling a text segment with a concept.

Figure 5.5 shows the average number of document segment per the document genre. Note that the numbers signify valid segments which have been annotated with a concept. In other words, the text segmentation subsystem may actually divide a document into more sections than this, however, a segment only becomes valid if human annotators find it to be a realistic section which can be assigned a concept from the list of concept already assigned to that document on the EurLex website.

The same topical segments for each document were fed into the developed system. The goal of the system is to quantify the meaning of these segments and for each, select the concept that is most semantically related. Using the manual annotation as *Gold Standard*, we compare the performance of the system with that of manual annotation using the popular information retrieval metrics: Precision and Recall. The documents were parsed and the text extracted. Usually, we remove the common headers which are found in all the documents. Each document is then passed through our text segmentation system which also does more text processing tasks. The segmented texts are passed to the semantic annotator which computes the semantic representation of each segment, this result into a single vector per segment. Similarly, all the concepts were expanded and their semantic representation computed based on the ESA method described in section (5.1). Each concept also corresponds to a vector. Both the segment and concept representation were indexed with *Faiss*[9].

The important task is to compare the vector of each concept to the vectors of all the indexed segment. The segment(s) with the vector that is most semantically similar to the vector of a concept when computed according to equation (2.10) is associated to that

[9] Available at https://github.com/facebookresearch/faiss. Faiss is a library for indexing vectors and performing efficient similarity search and clustering on the vectors.

Domain of documents	No of documents	Ave No. of Segment per Document
Transport Infrastructure	25	3
Inland Transport	25	5
Shipping	25	4
Air Transport	25	7

TABLE 5.5: Number of valid segment retained per document by human annotators.

Domain of documents	No of documents	Precision (%)	Recall(%)
Transport Infrastructure	25	0.74	0.77
Inland Transport	25	0.71	0.73
Shipping	25	0.72	0.74
Air Transport	25	0.68	0.71
Average Score	100	0.71	0.73

TABLE 5.6: Precision and Recall obtained with the Semantic Annotation task

concept. Evaluation is done by comparing the automatically generated concept-segment mapping to that of human-generated annotation.

5.5 Discussion

The Precision is the number of accurate tagging by the system in comparison to that of human annotators and it is calculated by the formula in equation (2.28). The Recall, on the other hand, is the number of accurate tagging made by the system, and it is calculated using the formula given in equation (2.29). Table 5.6 shows the results obtained under different categories of documents. We can see that we obtained the best Precision and Recall scores from the 'Transport Infrastructure' documents while the worst result comes from the 'Air Transport' category. Manual exploration of the documents reveal that the documents under this category with the best result are quite short (average of 3 pages) compare to those under Air Transport where the average number of ages was double that of the former. Overall, we obtained an average precision score of 0.71 and recall value of 0.73.

Table 5.7 shows our user evaluation of the text segmentation subsystem. We selected 150 segments generated automatically by the described system. The segments were derived from the 100 documents initially used in our experiment. The same human evaluators were used to provide judgment on the coherence of the segments. In doing this, they were provided with the original document from which each segment has been extracted. A judge examines whether the segment is plausible, considering the context of the boundary sentences to each segment. A judge does not need to worry about whether he can correctly associate a concept to that segment or not. In essence, we are only concerned about their decision concerning the segmentation accuracy or plausibility of the derived segments. Overall, 83% of the segments were accepted to be valid by human observers.

	No. Of Segments	Acceptable	Not Acceptable
Judge 1	**150**	126	24
Judge 2	**150**	132	18
Judge 3	**150**	136	14
Average		131 (%87)	19(%13)

TABLE 5.7: Human Evaluation of the Text Segmentation task

5.6 Semantic Annotation and Information Retrieval

A user who is searching for documents obviously want a simplified way to perform his/her search. An IR system, possibly a document management system may offer users the possibility to query the documents using a controlled list of concept. Now, a concept is an abstract term which amongst other characteristics, may not explicit appear in the body of documents. Even though a user knows the meaning of this concept, and has an understanding of the kind of documents he expect to retrieve, the algorithm on the other hand is completely obscured. Semantic annotation, or matching as technically proposed in this work does the job of obtaining the semantic representation of both the object (concept) and the subject (document/segment) and finding a correspondence between the matching objects and subjects.

With respect to IR,the goal is to match a query to a document or set of documents. Similarly, once the IR system understands that a particular concept semantically matches a particular document(s)/segment(s), the matching document(s)/segment(s) are retrieved. An interesting potential of our approach is its versatility, i.e., if all the documents in a collection are entered into such system, it can automatically segment the documents, and index each segment based on its conceptual representation as earlier described. Also, each concept is indexed based on its computed semantic representation as described in section (5.1). Users of the retrieval system are then given the option to select a concept (from a controlled list) that represents their information need. And the system retrieves all the relevant segments from different documents. It is also possible to query the system with a free-text query. Here, the system can represent the free text conceptually by expanding the constituent terms of the query by using the lexical and ESA representation as done in the case of the abstract concept. Then, a similarity between the semantic representation of the query and the representation of the indexed segments is carried out, and the top-ranked segments are returned to the user. Similarly, it is possible to obtain a conceptual representation for a whole document instead of segments, also computed with our concept representation techniques. These representations are then indexed as would the segment representation. In this case, instead of retrieving the segments, the system retrieves the full document.

5.7 Chapter Summary

In this chapter, we describe our conceptual passage retrieval system. The system makes use of a lot of self-contained components, such as the text segmentation system, semantic text similarity system, and the semantic annotation system. These components are plugged together to achieve the overall system as shown in Figure 3.1. We evaluated each of the sub-systems separately and benchmarked against some state-of-the-art systems. Our semantic annotation system incorporates knowledge from knowledge resources like the WordNet, as well as external sources like the Wikipedia and EUR-Lex texts. We achieved 71% and 73% precision and recall scores for the semantic annotation task. Our text segmentation system outperforms TopicTiling, a state-of-the-art text segmentation algorithm and has been validated by practitioners to derive meaningful segments from legal documents.

Part III

Electronic Discovery/ E-Discovery

Chapter 6

The Ensemble Relevance Matching Model

In this chapter, we introduce the important Neural Network algorithms which are important to our work. We also give a technical description of the methodology for the proposed Relevance matching model.

6.1 General Background

Given a query Q composed of terms q_1, q_2, q_3, ..., $\mathrm{q}_{|Q|}$, and a document D composed of terms d_1, d_2, d_3, ..., $\mathrm{d}_{|D|}$. Each term q or d is represented as a vector. Depending on the scoring function, a vector e could be a row of an embedding matrix $\mathbf{E} \in \mathbb{R}^{dim \times v}$, where dim is the dimension of each row of the matrix and v is the size of the vocabulary. The embedding matrix could be generated through a latent semantic approach (Deerwester et al., 1990) or a distributed approach (Mikolov et al., 2013b). A vector could also be represented as a *one-hot* encoding of each word. In our work, this is useful when using the traditional BOW approach for ranking function. Our goal is to compute a set of scores F = f_1, f_2, ..., f_N. Where N is the number of features being combined. Each f_i(r(,)) is a feature extraction layer that takes as input both the document and the query and uses a scoring function r to approximate a score, i.e., f(r(Q , D)). As we will show, r could be a simple *cosine* function, a fully connected *MLP* , or any other neural network that outputs a matching score given two input representations. Below, we describe two Neural Network algorithms which are essential components of our model, i.e., the LSTM and CNN. Also, we give some details about our methods for encoding the input terms and obtaining semantic representation of the terms with the Neural Network components.

6.2 Sentence level feature extraction with Long Short-Term Memory Neural Network

Recurrent Neural Networks (RNNs) have connections that have loops, adding feedback and memory to the networks over time. This memory allows this type of network to learn and generalize across sequences of inputs rather than individual patterns. LSTM Networks (Hochreiter and Schmidhuber, 1997) is a special type of RNNs and are trained using backpropagation through time, thus overcoming the vanishing gradient problem. LSTM networks have memory blocks that are connected as layers, the block contains gates that manage the block's state and output. These gates are the *input* gates which decides the values from the input to update the memory state, the *forget* gates which decides what information to discard from the unit and the *output* gates which decides what to output based on the input and the memory of the unit. LSTMs are thus able to memorize information over a long time-steps since this information is stored in a recurrent hidden vector which is dependent on the immediate previous hidden vector. A unit operates upon an input sequence and each gate within a unit uses the sigmoid activation function to control whether they are triggered or not, making the change of state and addition of information flowing through the unit conditional.

At each time step t, let an LSTM unit be a collection of vectors in $\mathbb{R}^d$ where d is the memory dimension: an *input gate* i_t, a *forget gate* f_t, an *output gate* o_t, a *memory cell* c_t and a *hidden state* h_t. The state of any gate can either be open or closed, represented as [0,1]. The LSTM transition can be represented by the following equations (x_t is the input vector at time step t, σ represents sigmoid activation function and $\odot$ the elementwise multiplication. The u_t is a tanh layer which creates a vector of new candidate values that could be added to the state) :

$$\begin{aligned}
i_t &= \sigma\Big(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)}\Big),\\
f_t &= \sigma\Big(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)}\Big),\\
o_t &= \sigma\Big(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)}\Big),\\
u_t &= \tanh\Big(W^{(u)}x_t + U^{(u)}h_{t-1} + b^{(u)}\Big),\\
c_t &= i_t \odot u_t + f_t \odot c_{t-1},\\
h_t &= o_t \odot \tanh c_t
\end{aligned} \tag{6.1}$$

6.3 Attention Layer

We introduce an attention layer in order to obtain a more informative representation of the query and document terms during encoding. Attention is a way to focus intensely on some important parts of an input, and it has been used extensively for some language modeling tasks (Bahdanau, Cho, and Bengio, 2014; Parikh et al., 2016). Essentially, it is able to identify the parts of a text that are most important to the overall meaning of the text. Specifically, such important words can now be aggregated to compose the meaning of that text. Assume that h$_i$ is the annotation of the i-*th* word from the word encoding, the annotation h$_i$ is passed through a single layer MLP to get a hidden representation u$_i$ (see equation 6.2). The contribution of a word to the overall meaning of a text can be computed based on how similar the hidden representation u$_i$ is to a word level context vector u$_t$. The context vector is analogous to a container with a fixed value and we want to measure how close our hidden representation is to the fixed value. The context vector could be randomly initialized parameter and jointly learned during the training, otherwise, the hidden annotation of the previous input is used, in this case, the context vector at time-step *t=1* will be randomly initialized. We then obtain a normalized importance weight α_i as shown in equation 6.3. This is computed with a softmax function. The weights from the attention vector α_i sum up to 1, and are used to compute a weighted average of the word annotation (last hidden layers) generated after processing each of the input words. In this scenario, h$_s$ in equation 6.4 becomes the sentence representation instead of using the final hidden state from the dual word encoding.

$$u_i = \tanh(W_p h_i + b_p) \tag{6.2}$$

$$\alpha_i = \frac{\exp(u_i^M u_t)}{\sum_i \exp(u_i^M u_t)} \tag{6.3}$$

$$h_s = \sum_i \alpha_i h_i \tag{6.4}$$

6.4 Word Encoding

Each input word to a Neural Network input layer has to be represented with a descriptive vector that captures the semantics of the word. Here, we represent each query or document word with a d-dimensional vector, where the vectors are obtained from a word embedding matrix. Assume that each of the inputs contain words x$_i$, x$_{i+1}$, x$_{i+2}$, x$_{i+3}$, ..., x$_n$. We associate each word w in our vocabulary V with a vector representation x$_w \in \mathbb{R}^d$. Each x$_w$ is of dimension $d \times |V|$ of the word embedding matrix W$_e$, where $|V|$ is the size of the vocabulary and d is the dimension of the word embedding vector. Generally, we make use of the 300-dimensional *GloVe* vectors, obtained from 840 billion words Pennington, Socher, and Manning, 2014. It is also possible to train an embedding algorithm like *Word2Vec* (Mikolov et al., 2013b) on the document collection. However, we observed an improved performance during training when using the pre-trained vectors. A Bi-directional LSTM is used in order to obtain contextual information between the words. A Bi-directional LSTM is essentially composed of two LSTMs, one capturing information in one direction from the first time step to the last time-step while the other captures information from the last time-step to the first. The outputs of the two LSTMs are then combined to obtain a final representation which summarizes the information of the whole sentence. Equations (6.5) and (6.6) describes this computation.

$$\overrightarrow{h_i^p} = LSTM(\overrightarrow{h_{i-1}^p}, P_i), \quad i \in [1, ..., M]$$

$$\overleftarrow{h_i^p} = LSTM(\overleftarrow{h_{i-1}^p}, P_i), \quad i \in [M, ..., 1] \tag{6.5}$$

$$h_i = [\overrightarrow{h_i^p}; \overleftarrow{h_i^p}] \tag{6.6}$$

Typically, when using an ordinary LSTM or BiLSTM to encode the words in a sentence, the whole sentence representation can be obtained as the final hidden state of the last word or time-step. We encode and obtain the sentence representation of each input text using the following equation:

$$\vec{h_a} = Attention_{intra}(BiLSTM(\vec{A})) \tag{6.7}$$

where $Attention_{intra}$(A) is a function for obtaining attention weighted representation of an input A according to equations 6.2 to 6.4, and BiLSTM(A) is a BiLSTM encoder obtained with equations 6.5 and 6.6. The $Attention_{intra}$ function uses the annotations, i.e., the internal representation of each word from the BiLSTM. Similarly, instead of using a single d-dimensional vector which is obtained as the average of all attention vector (α_i in equation 6.3) weighted time-steps of the sentence (see equation 6.4), we instead obtain the weighted representation for each time-step, such that h$_p$, h$_q$, and h$_a \in \mathbb{R}^{d \times n}$ (i.e., we drop the sum in equation 6.4).

6.5 Hierarchical Attention for Input Interaction

We introduce two forms of attention, i,e, the intra-sentence attention and the inter-sentence attention. This results into an hierarchical attention which induces necessary interaction between the inputs. Given two inputs Q and D which are the query and document terms. This intra-attention works by focusing on the important words within Q and D and it is computed according to the equation (6.2) - (6.4). Secondly, the inter-attention creates an interaction between the two inputs by looking at the important words in the query in the context of the terms in the document, and vice versa. Specifically, what this means is that the model takes in the intermediate representations of input Q and D according to the equations 6.7. The model then uses the intermediate representation of Q to create another attention (*inter*) which is conditioned at each time steps of D. We then use a matching-function which is similar to the one proposed by Wang et. al., (Wang, Hamza, and Florian, 2017). The matching function creates a similarity interaction between two texts, i.e., from one text to another text. Also, we utilized the conventional cosine similarity without an additional trainable parameter. The matching function works as explained below.

$$\overrightarrow{match_i}^{forward} = sim(\overrightarrow{h_i^Q}, \overrightarrow{h_i^D}) \tag{6.8}$$

$$\overleftarrow{match_i}^{backward} = sim(\overrightarrow{h_i^D}, \overrightarrow{h_i^Q}) \tag{6.9}$$

$$sim = cos(V1, V2) \tag{6.10}$$

Given two inputs Q and D, we represent an interaction (Q→D) by a forward pass and interaction (D→Q) by the backward pass. In the forward pass (*see* equation 6.8), we compare the time-step from the last hidden state of Q to every time-steps of D. Similarly, in the backward pass (*see* equation 6.9), the computation is done in a similar way. We compare the time-step from the last hidden state of D to each of the time-steps in Q . For both forward and backward passes, the comparison is done by obtaining how similar the two vectors are, using the cosine function in equation (6.10). The cosine function makes use of the cosine similarity formula in equation 2.10. This matching function creates a form of interconnection from one-time-step to every other time-steps, thus yielding two vectors of interaction signals. In the original full-matching method of (Wang, Hamza, and Florian, 2017), they compared each time-step from one text to every time-step in the other text. Furthermore, the comparison is done with a Bi-LSTM which makes the approach further computationally expensive. Here, we only compare the sentence representation of one sentence with each word in the other sentence and vice-versa. Also, for simplicity, we use the hidden state from the last time-step of a text as its encoding representation.

6.6 Interaction Vector Normalization

The interaction vectors obtained may have variable size, depending on the number of time-steps in D or Q. This can be normalized by introducing a matching histogram. The matching histogram can group the signals according to their strengths. The signals are similarity scores which range between [-1, 1], thus it is possible to introduce a fixed-size ordered bins such that a fixed size interaction vector which contains counts of local signals in each bin for both D and Q as done by (Guo et al., 2016). We utilize a bin size of 0.2 such that we derive ten bins { [-1, -0.8], [-0.8, -0.6], [-0.6, -0.4], [-0.4, -0.2], [-0.2, -0], [0, 0.2], [0.2, 0.4], [0.4, 0.6], [0.6, 0.8], [0.8, 1.0] }. If for instance, the interaction vectors for D (Italy, is, the, home, of, pizza, and, pasta) and Q (Italians, love, pizza) respectively are [0.5, 0.1, 0.6] and [0.4, 0.1, 0.1, 0.2, 0.1, 0.6, 0.1, 0.3]. We can see that the interaction vector of Q has three signals while that of D has eight signals which correspond to the number of the time-steps of both Q and D respectively. By counting the number of signals for each local bin, we generate a uniform-sized vectors as [0, 0, 0 , 0, 0, 1, 0, 2, 0, 0] and [0, 0, 0 , 0, 0, 5, 2, 1, 0, 0] for P and Q respectively. An alternative to the matching histogram is to introduce a maxpooling function in order to select the best signals from each interaction vector.

Once we obtain the uniform-sized interaction vectors, we introduce a *Merge* layer where the two vectors are concatenated. The resulting vector is then passed to a fully connected Multilayer Perceptron (MLP) network.

6.7 Sentence level feature extraction with Convolutional Neural Network (CNN)

Here, we apply CNN to extract and compose important features from the query and document representations, which can then be used for necessary classification. There are important motivations for using CNN for sentence modeling tasks, i.e., it allows parameter sharing, the use of convolution filter enables the neural network to induce interaction within a small window of words rather than over the whole sequences of words in the sentence. Several filters also extract different features across these windows. Lastly, with the k-MaxPooling feature, it is possible to select the best features from those returned by the individual filter. These qualities of CNN have contributed to its success in many NLP tasks. Our CNN architecture is essentially similar to the one used by Kim (Yoon, 2014) for sentence classification. Let h(t) $\in \mathbb{R}^d$ be the d-dimensional vector corresponding to the t-th time-step in Q and D. We pad each input representation up to a fixed length. Usually, the fixed length chosen should reflect the maximum sequence length in the training sample. Assume a padding size *n*, a time-step *t* and a concatenation operator $\oplus$, after padding, we concatenate each element in the sentence representation such that each sentence representation is as shown in equation (6.11). Here, h(t) represents the hidden state at a particular time-step *t*.

$$h(t:n) = h(t) \oplus h(t+1) \oplus h(t+2) \oplus ... \oplus h(t+n-1), \tag{6.11}$$

Assume a window size w which is a local receptive field of a convolution. Also, let h(t:t+j) be the concatenation of time-steps h(t), h(t+1), h(t+2),...., h(t+j). We can apply a convolution filter $\mathbf{F} \in \mathbb{R}^{w \times d}$ to the concatenated sequence in each sentence representation. Each filter captures a window of w time-steps in the sentence representation in order to produce a new feature which is applied to a window of h words to produce a new feature. By applying a filter of size w to the receptive field h(t:t+w-1), we obtain a local feature c(t) as shown in equation (6.12).

$$c(t) = \tanh(\mathbf{F} \bullet \mathbf{h}(t : t + w - 1) + b) \tag{6.12}$$

where b $\in \mathbb{R}$ is a bias vector, and $\tanh$, the hyperbolic tangent is a non-linear function. Other non-linear functions like the Rectified Linear Unit (ReLU) or the logistic sigmoid are a possibility. In reality, the filter **F** is applied to multiple window of time-steps e.g., {h(t:w), h(t+1:w+1), ..., h(n-w+1:n)} in order to obtain a *feature map c* as shown in equation (6.13).

$$c = [c_1, c_2, ..., c_{n-w+1}] \tag{6.13}$$

where c $\in \mathbb{R}^{n-w+1}$. Once the feature map is obtained, a *k-MaxPooling* operator can be applied to extract the *k* strongest features from a *feature map* as shown in equation (6.14). Literally, what the operator does is to take the features with the highest values and thus obviates the variation in length of the input sequence. The value chosen as *k* is a matter of choice but generally, even though setting *k* to be higher than one introduces more parameters, it does not always lead to an increase in performance. In our experiment, we set $k = 1$. Considering that a model in practice uses multiple (N) filters, there will be $k \times N$ dimensional output vectors, e.g., **Z** = [c$_{max}$(1), c$_{max}$(2), c$_{max}$(3), ..., c$_{max}$(N)]. In particular, having several parallel filters with each extracting the strongest features from a collection of receptive fields is the selling point of the CNN. If desired, Z may further be propagated as a feature into other neural network components or a fully connected MLP layer. Figure 6.1 shows a high-level view of the LSTM-CNN architecture where the weights and parameters of the CNN are jointly shared, hence a local interaction is created between the two vector representations.

$$c_{max} = max\{c\} \tag{6.14}$$

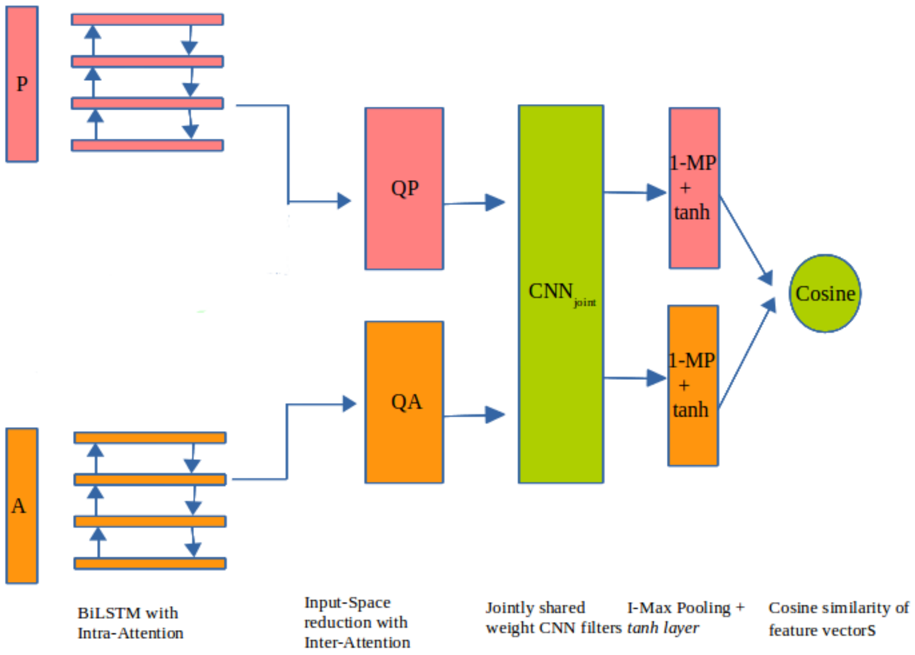

P
A
QP
QA
CNN$_{joint}$
1-MP + tanh
1-MP + tanh
Cosine
BiLSTM with Intra-Attention
Input-Space reduction with Inter-Attention
Jointly shared weight CNN filters
I-Max Pooling + *tanh layer*
Cosine similarity of feature vectorS

FIGURE 6.1: LSTM-CNN with jointly shared convolutional layer parameters

We describe each feature extraction layer below:

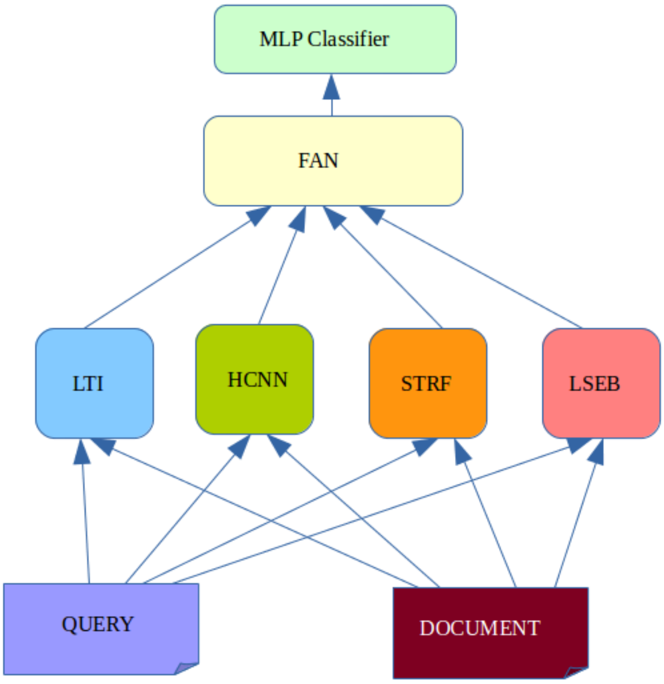


FIGURE 6.2: Schematic Representation of the Ensemble Relevance Model.

6.7.1 Semantic Text Representation Feature (STRF)

Here we use Convolutional Neural Network (CNN) to generate a semantic representation of both the query and the document. Basically, our approach is the same as described in section 6.7. The query and document terms are encoded with vectors from a distributed word embedding matrix. Here, we utilize the GloVe vectors. The encoded terms are then passed through a CNN as described in that section. The two representations (i.e., document and query) have a jointly shared Convolutional layer parameters and are then passed through a feed-forward Neural Network. Here, the scoring function r is the cosine similarity. This feature extraction layer is very similar to *Arc-I* (Hu et al., 2014) and the CNN text classification model of (Yoon, 2014). Schematically, it resembles the model in figure 6.1 without considering the input layer.

6.7.2 Local Query-Document Term Interaction (LTI)

This approach is very simple, we encode the query and document terms with the word embedding Matrix $\boldsymbol{E}$. Subsequently, we pass the embedded query and document terms into a separate LSTM neural network. This also follows the approach of (Palangi et al., 2016) where the LSTM was combined with a vanilla RNN. The output of this stage is the hidden state representation of the embedded terms in both the query and the document. The internal representations for each of Q and D can be viewed as a matrix whose row is the hidden state representation for each constituent term. Next, we induce a word-word similarity interaction between the two internal state representations as described in section 6.5. Using equations 6.8, 6.9, and 6.10 , we compare each hidden state representation of each term in Q with every term in D. In essence, we obtain a matrix $\boldsymbol{M} \in \mathbb{D}^{|Q| \times |D|}$ whose i-th row contains the similarity scores between the i-th time-step of Q and the j-th time step of D (j = 1,2,...,∣D∣). What this feature does is to identify the matching points between the query terms and the document terms. Next, we pass the similarity matrix $\boldsymbol{M}$ through a Convolutional layer with c filters. All the computation in the Convolutional layer, as well as the parameters, are as defined in section 6.7, such that the output of each filter is computed according to equation (6.12). Finally, we pass the output from this layer through a *MLP* with a $\tanh$ layer. Here, the MLP is used as the scoring function and $\tanh$ ensures that a continuous value output which captures the similarity interaction between the terms is produced. This approach is closely similar to the local model of the DUET architecture (Mitra, Diaz, and Craswell, 2017) where the authors used the alignment between the one-hot encoding of the query and document terms to build a local interaction matrix. This is also related to the matching by an *Indicator function* as proposed in (Guo et al., 2016), where an exact matching position in the interaction matrix is signaled by a *1* and the points where there is no matching takes a value *0*. However, unlike in their work, our approach captures not only an *exact matching* interaction but also a semantic matching via the distributed representation.

6.7.3 Position-Aware Hierarchical Convolution Query-Document Interaction (HCNN)

We include a hierarchical Convolution interaction scoring following Arc-II (Hu et al., 2014). The key difference is that instead of using the multi-layer convolution on the input representations of both the query and the document, our approach uses an interaction similarity matrix. As opined by (Guo et al., 2016), strictly performing a semantic matching does not work well in a situation where the inputs are non-homogeneous. Furthermore, in IR, the length of the document is not commensurate with that of a query, by obtaining a deeper semantic representation for the query and the document, and matching using these representations, the model loses the focus on the position where important matching occurs between the query and the document terms. This position-awareness really matters in relevance matching. Arc-II is suitable for tasks like text similarity and other natural language inference tasks but may not scale well in a large scale information retrieval settings. Here, we follow the approach described in the section above to generate a similarity score matrix, but we include a multi-level convolution for locating several positions of matching. What this means is that we care less about obtaining an overall sentential representation of the query terms because they lack syntactic or grammatical cohesion. Instead, we focus on the different parts of the document that the query term matches. A schematic representation of this approach is shown in figure 6.3.

6.7.4 Latent Semantic Embedding and BOW Feature (LSEB)

Despite the proposal of more sophisticated algorithms by researchers in IR over the years, the simple approach that is yet to go away is the count-based approach to IR. The reason is simple, their simplicity does not reflect in the undeniability of their performance. One extension of the count-based approach is to weigh terms with TFIDF (Salton and Buckley, 1988). The BM25 (Robertson et al., 1995; Robertson and Zaragoza, 2009) weight scoring is a probabilistic extension of the TFIDF approach. The Latent Semantic Analysis (LSA) builds on the simple term-weighing approach by capturing the term-document relationship in a collection. The motivations and the methods for using these algorithms are contained in chapter 2. In this work, we utilize these three approaches for generating a matching score for each query-document pair as explained below.

- TFIDF scoring: First, all the terms in both query and document are count-vectorized and then weighted with TFIDF. We obtain two TFIDF-weighted vectors, one for the query and the other for the document. These vectors are of the same dimension. We simply pass a concatenation of the two vectors through a MLP which learns to predict a similarity score between the two vectors.
- BM25-scoring: Here, we use the BM25 algorithm to generate a score for the query. The ranking score generated is normalized and scaled to fall between *-1* and *1*. We pass the TFIDF-weighted vectors as above through a MLP with a tanh layer to predict a ranking score.
- LSA-scoring: We trained the LSA algorithm on the full document collection. A vector is generated for the query and the document based on the LSA model. We pass a concatenation of the two vectors through a MLP with a tanh layer to predict a ranking score.

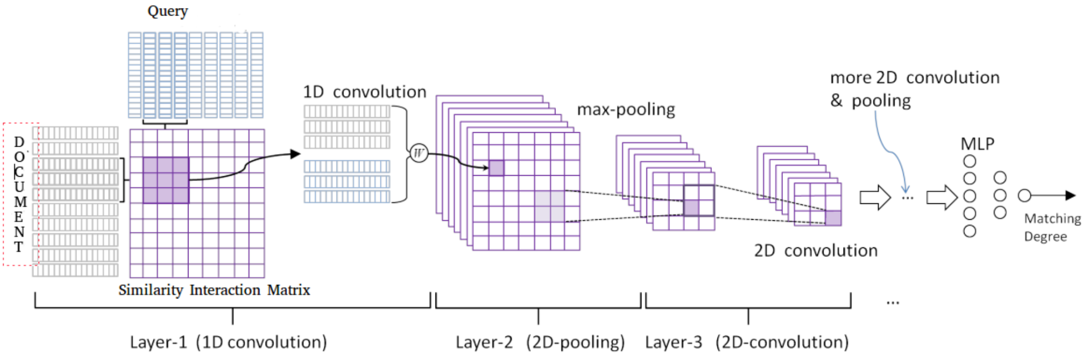


FIGURE 6.3: Hierarchical Convolution on Query-Document Similarity Interaction.
(The model diagram is partly adapted from (Hu et al., 2014))

6.8 The Feature Aggregating Network (FAN)

Each of the models described can be seen as a feature extractor[1] for the final ranking model. The final ranking model incorporates the ranking scores from each of these models and aggregate a final ranking score. We introduce an aggregating network similar to the gating network of (Guo et al., 2016). This network combines individual matching score into a final score by passing each score through a $\tanh$ layer and weigh with each output weighted with a softmax function. Assume an initial feature scorer $\mathrm{f}^{(0)}$, e.g., this could be the *semantic text representation* feature scorer which was first described above or any other subsequent ones. We can represent the aggregating network as below.

$$f^{(0)} = r(q_i \otimes d_j), \quad i = 1, 2, ..., |Q|; J = 1, 2, ..., |D|$$

$$f^{(l)} = \tanh(W^{(l)} f^{(l-1)} + b^{(l)}), \quad l = 1, 2, ..., N$$

$$Score_{final} = \sum_{l=0}^{N-1} \sigma f^{(l)} \tag{6.15}$$

Where $\otimes$ represents the interaction between the query and document term. This could be a full matching interaction or just the BOW similarity. *r* signifies the matching function, $\mathrm{f}^{(l)}$ denote the feature score from a single model identified by *l*, and lastly, $\mathrm{W}^{(l)}$ and $\mathrm{b}^{(l)}$ are the weight matrix and the bias vector for an *l*-th feature scorer.

6.9 Training

For the distributed word embedding, we utilize the GloVe *840b, 300D* vector which has been used in the other experiments. During encoding, out of vocabulary words are randomly assigned a vector sampled between -0.25 and 0.25. For the latent semantic analysis, we use Gensim's [2] implementation of LSA on the whole TREC Legal 2010 collection. We also make use of the BM25 scoring function of the Gensim. The model was trained with hinge loss (see equation 6.16). The training description and parameters are set similar to the ones described in section **??** of chapter **??**. Each training sample contains a query, a responsive document for that query and a non-responsive document for the query. Given a *train* sample containing a Query q, a Responsive document d^+, and a non-responsive document d^-, represented as a triple- (q, d^+, d^-), the pairwise ranking is such that the matching score of (q, d^+) exceeds that of (q, d^-). In essence, the goal is to create a hard margin between d^+ and d^- and the loss function is defined as

$$L(q, d^+, d^-, \theta) = \max\{0, 1 - Score_{final}(q, d^+) + Score_{final}(q, d^-)\} \tag{6.16}$$

[1] This is probably an abuse of terminology. What we mean by feature is the ranking score generated by the individual model

[2] https://radimrehurek.com/gensim/

Here, $\text{Score}_{final}(\ ,\ )$ is the predicted matching score for two inputs. The parameters of the network are represented by θ. The network is trained via back propagation.

6.10 RFP Topic Reformulation and Query Expansion

One of the ways by which E-Discovery is different from the general IR is the way in which the information need is presented. Here, the RFP is coded by the legal team, based on the complaint received from the court. As shown in figure 4.3, the main topic says

> *All documents or communications that describe, discuss, refer to, report on, or relate to the design, development, operation, or marketing of enrononline, or any other online service offered, provided, or used by the Company (or any of its subsidiaries, predecessors, or successors-in-interest), for the purchase, sale, trading, or exchange of financial or other instruments or products, including but not limited to, derivative instruments, commodities, futures, and swaps.*

However, only a few parts of the text carries the important message or the information need. For example, the part-

"All documents or communications that describe, discuss, refer to, report on, or relate to the."

is not so useful. Therefore, as much as possible, it is necessary to weed out the redundant part. In this thesis, we manually analyze the topics in order to reformulate it into a suitable query. For example, using the above topic as an example, we will retain the following part:

> *design, development, operation, or marketing of enrononline, or any other online service offered, provided, or used by the Company (or any of its subsidiaries, predecessors, or successors-in-interest), for the purchase, sale, trading, or exchange of financial or other instruments or products, including but not limited to, derivative instruments, commodities, futures, and swaps.*

Usually, lawyers go through the topic reformulation process in an iterative way. After reformulation, they query the collection with the reformulated topics and observe the performance on a selected sample. If the precision or recall on the sample is poor, they may decide to include other terms or reformulate the query in order to improve the performance of the system. This may be likened to relevance feedback.

After reformulation, it is important to expand the query so as to overcome a language variability issue like synonymy. Expanding the query allows for the inclusion of important terms which may be missing in the RFP but is pertinent to a good retrieval performance. There are many techniques already proposed for query expansion and we have discussed several of them in chapter 2. The easiest way is to use a knowledge graph like the WordNet to retrieve the best synonyms of each term in the reformulated query.

While this may work in theory, WordNet does not contain all the words one could ever encounter. For example, a word like *enrononline* would be missing in the WordNet. A practical solution is to make use of a broader and bigger knowledge base which specifically captures semantic relatedness information. We have already explained what word embedding is in chapter 2, in fact, most of our solutions in this thesis have strongly employed word embedding.

A word embedding is like a matrix whose rows are the vectors that carry descriptive information about a word. In particular, because they are obtained from a distributed analysis of a corpus, they not only incorporate semantic similarity but much more importantly, they capture relatedness. We believe relatedness is exactly what is essential in expanding queries for E-Discovery. For example, when we check for the most similar terms to the word *'Enron'* using a Word2Vec pre-trained word embedding[3], we obtained the words "`Soros, Scandal, Martha Stewart, WallStreet, Gordon Brown, and Automatic fuel injected`" in the top *6* most similar words. Now, this model may not have been trained on many documents related to *Enrononline*. In essence, a model trained on a bigger corpus (e.g., Wikipedia) or even the GloVe vectors trained on *840* billion words would have amassed an incredible amount of knowledge and it is even better if trained on the entire text collection. In summary, we borrowed our concept expansion approach described in chapter 5 which makes use of a knowledge graph (WordNet), an embedding model (GloVe vectors), and an explicit semantic analysis (ESA) of the Wikipedia. This approach is well described and the reader is referred to section 5.1. Moreover, this method already yielded a strong performance during manual inspection while running our experiments (see details in chapter 5). Assume that each of the important words in the reformulated query is viewed as a concept, the expanded terms for that word is obtained with equation 5.1. A combination of all the expanded terms as computed with equation 5.1 gives the new query terms which we make use of as the query while training our model.

6.11 Experiment

Here, we describe the experiments conducted using our Relevance-Matching classifier that takes a document and a query, and determines whether the document is relevant or not to the query.

The TREC conference provides the best opportunity for benchmarking systems for large-scale information retrieval. The authors in (Voorhees and Harman, 2005) already articulated the fundamentals of the TREC evaluation. Our experiment makes use of the TREC Legal track data. However, the track has been discontinued since 2011, probably due to the sheer amount of efforts that is required in order to get a valid set of relevance judgment. Unlike in other IR tracks, the legal track requires the use of experts and at least, law

[3] GoogleNews-vectors-negative300

students to assess document for relevance judgment. The overview of past competition contains detail description of these tasks (e.g., see (Cormack et al., 2010) for 2010). There are two tasks proposed for the 2010 TREC legal track, the *learning task*, and the *interactive task*. In this thesis, we are majorly concerned about the interactive task, however, we report our experiment for the learning task.

6.11.1 The Interactive Task

According to the organizer, the Interactive task fully models the conditions and objectives of a search for documents that are responsive to a production request served during the discovery phase of a civil lawsuit. Teams are expected to produce a binary output (Responsive or Non-responsive) to each query-document pair in the collection. This task better mimics a classification task. Training a NN algorithm requires a significant amount of data, and luckily, the TREC legal track provides a sizable amount of training data. Moreover, more relevance judgment is made available which helps in better training our classifier. The full description of this task and some other details can be found in (Cormack et al., 2010). For our experiment, we downloaded the TREC 2010 legal track data (edrmv2txt-v2) which contains the email used in the Enron civil case and the TREC Legal 2009 data. The 2009 data is a collection of emails that had been produced by Enron in response to the requests from Federal Energy Regulatory Commission (FERC) (Hedin et al., 2009). The messages contain attachments which exemplify a real-world E-Discovery text collection. The emails belong to 150 employees of Enron Corporation and were created between 1998 and 2002. In total, there are 569,034 distinct messages embedding some 278,757 attachments. The total text collection stands at 847,791 documents (when parent emails and attachments are counted separately). The following topics were made available for participants of 2010 interactive task.

1. **Topic 301**.

   > *All documents or communications that describe, discuss, refer to, report on, or relate to onshore or offshore oil and gas drilling or extraction activities, whether past, present or future, actual, anticipated, possible or potential, including, but not limited to, all business and other plans relating thereto, all anticipated revenues therefrom, and all risk calculations or risk management analyses in connection therewith.*

2. **Topic 302**.

   > *All documents or communications that describe, discuss, refer to, report on, or relate to actual, anticipated, possible or potential responses to oil and gas spills, blowouts or releases, or pipeline eruptions, whether past, present or future, including, but not limited to, any assessment, evaluation, remediation or repair*

> *activities, contingency plans and/or environmental disaster, recovery or clean-up efforts.*

3. **Topic 303**.

   > *All documents or communications that describe, discuss, refer to, report on, or relate to activities, plans or efforts (whether past, present or future) aimed, intended or directed at lobbying public or other officials regarding any actual, pending, anticipated, possible or potential legislation, including but not limited to, activities aimed, intended or directed at influencing or affecting any actual, pending, anticipated, possible or potential rule, regulation, standard, policy, law or amendment thereto.*

4. **Topic 304**.
   "Should Defendants choose to withhold from production any documents or communications in the TREC Legal Track Enron Collection on the basis of a claim of privilege, attorney work-product, or any other applicable protection, they should identify all such documents or communications."

In particular, Topic 304 includes a privilege review, i.e., participants should determine whether a responsive document for the topic contains any privilege information. This task specifically ensures that the documents that have been marked to be relevant are producible.

There are two different ways of assessing a collection, especially an email collection like the Enron collection used in both 2009 and 2010 tasks. For instance, the collection could be assessed for effectiveness at the message level (i.e., treat the parent email together with all of its attachments as the unit of assessment) or at the document level (i.e., treat each of the components of an email message (the parent email and each child attachment) as a distinct unit of assessment. In the Interactive task for TREC 2010, participants are expected to submit their assessment at the *document level*. The assessment is then performed based on the following rules given by the organizers:

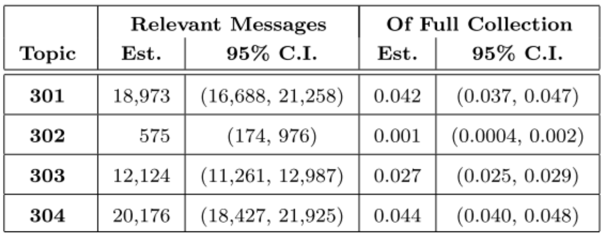
- A parent email should be deemed relevant either if in itself, it has a content that meets the definition of relevance, or if any of its attachments meet that definition; contextual information contained in all components of the email message should be taken into account in determining relevance.

- An email attachment should be deemed relevant if it has content that meets the Topic Authority's definition of relevance; in making this determination, contextual information contained in associated documents (parent email or sibling attachments) should be taken into account.

- A message will count as relevant if at least one of its component documents (parent email or attachments) has been found relevant.

- For purposes of scoring, the primary level is the message-level; document-level analysis is on between documents reviewed and supplementary. By contrast, the Learning task reports only document-level analysis.

For each topic, participants are expected to submit a classification result for all the documents in the collection. We trained our model on the TREC 2009 data, using the relevance judgments provided for the Batch and the Interactive tasks. Note that the dataset for 2009 Batch task is different from the one for the Interactive task. While the Interactive task uses a version of Enron email, the Batch task uses the IIT Complex Document Information Processing Test Collection, version 1.0[4]. The relevance judgment from the Batch task contains a total of *20,683* samples, divided into 10 topics. For topics 7 and 51, included are the judgments from the 2006 Ad Hoc task and the residual judgments from the 2007 Interactive and Relevance Feedback task. For topics 80 and 89, included are the judgments from the 2007 Ad Hoc task and the residual judgments from the 2008 Relevance Feedback task. For topics 102, 103 and 104, included are the post-adjudication judgments from the 2008 Interactive task. For topics 105, 138 and 145, included are the judgments from the 2008 Ad-Hoc task. Refer to table 6.1 for the summary of Batch 2009 relevance judgment. There are four relevance judgments for the Interactive task, i.e., the pre-adjudication judgment for message-based assessment and the one for document level assessment, as well as the post-adjudication judgment for message and the document. We only utilize the post-adjudication judgment for both message and document. The post-adjudication judgment for the message-level assessment contains *29,206* samples while that of the document-level assessment is *24,206*. Seven topics were provided in the relevance judgment, i.e., topics 201-207. Altogether, the relevance judgment from TREC 2009 data contains *74,095* samples. Similar to the Batch 2009 interactive task, the organizers provided 4 sets of relevance judgments for the 2010 Interactive task. Also, we are more interested in post-adjudication judgments. The post-adjudication judgment for message contains 25,507 relevance judgments while the post-adjudication judgment for document contains 46,331 relevance judgments. The estimated yield computed by the organizers is shown in figure 6.4. In total, there are *71,838* relevance judgments for the 2010 interactive task which have been used for the evaluation of our model.



	Relevant Messages		Of Full Collection	
Topic	**Est.**	**95% C.I.**	**Est.**	**95% C.I.**
301	18,973	(16,688, 21,258)	0.042	(0.037, 0.047)
302	575	(174, 976)	0.001	(0.0004, 0.002)
303	12,124	(11,261, 12,987)	0.027	(0.025, 0.029)
304	20,176	(18,427, 21,925)	0.044	(0.040, 0.048)

FIGURE 6.4: Estimated yields (C.I.=Confidence Interval) for the Interactive task 2010

[4] https://ir.nist.gov/cdip/

Topic	# Responsive	# Non-Responsive	Total
102	1548	2887	4500
103	2981	3440	6500
104	92	2391	2500
105	115	540	701
138	63	472	600
145	52	297	499
51	788	1259	1361
7	307	951	1269
80	721	1139	1879
89	164	607	874

TABLE 6.1: Summary of Batch task 2009 Relevance Judgment

We only have *74,095* relevance judgments from the 2009 data. In order to train the system, we have to create a training sample which contains triples of the topic; a responsive document for that topic; and a non-responsive document for that topic, i.e., in the format (q, d^+, d^-). The responsive document for a topic is its positive sample, while the non-responsive document is a negative sample. Also, in order to populate the train set, we created some synthetic positive and negative classes by randomly sampling 3 negative samples from the pool of the documents that are paired with another topic. In total, our training set consists of 142,933 samples which are triples of topic/query; positive document; and negative document. Note that usually during the competition, participants in the Interactive task are expected to judge every document in the collection for relevance with respect to each topic, and the organizers would then use a sampling method to select a strata for each topic. This leads to the selection of a subset of the sample to be used for evaluation by the assessors. Since the competition has been discontinued, we have no access to how this sampling is made, however, our assumption is that it has been coded into the relevance judgment bundled with the dataset. Apart from this uncertainty, we can have an effective comparison with the result of the participants as discussed in (Cormack et al., 2010). Table 6.2 shows the result obtained from our experiment. Furthermore, we compare our result with the submitted systems for TREC legal track 2010 interactive task. This comparison is displayed in the table 6.4. We observe a strong performance from our system,the best performance (F_1) was on *Topic 303*. We can see that this is consistent with the result of other participants. The model obtains a good precision score under Topics *301* and *302*. Since E-Discovery is a recall-oriented information retrieval task, it is satisfying to see good recall score especially for topic *303* and *304*. Overall, we notice that there is a good balance between the precision and the recall, which is important for E-Discovery because an omission of a relevant document could be costly than mere retrieval of non-relevant ones. Finally, we can see that our model outperformed the compared systems. This is particularly interesting considering that we do not have access to or make use of any *Topic Authority*'s advise or expertise unlike the real participant

Topic	Recall	Precision	F_1
301	.391	**.881**	.541
302	.295	.820	.433
303	**.815**	.770	**.791**
304	.736	.425	.538

TABLE 6.2: Evaluation Of Our Model On The 2010 TREC Legal Track Interactive Task Relevance Judgment
. **NB**: Topic 304 is a privilege review task.

in the task.

6.11.2 The Learning Task

It is possible that a classifier outputs a binary decision -*Responsive*/*Not-Responsive*, for example as done in the Interactive task. Even though a classification sorting like that suffices for the E-Discovery task, often times, it is not sufficient just to know that a document is relevant, it may be necessary to know how certain the classifier is in deciding that the document is relevant. Furthermore, imagine if the whole collection is made up of *6* million documents out of which *2* million documents have been marked as relevant by the classifier, we know that the classifier would have made some errors, however subtle.

An ideal thing would be to output a score or probability for each document, if we sort these probabilities in descending order, we can select just the documents at a specific cut off, say just the first 100,000 documents with the highest probabilities. Put in another way, if I search the Internet with Google using some keywords, Google would give me a sorted list of relevant pages, usually 1-20 items per page. Does it make any sense to start checking the pages at the bottom of the page? The answer is no, and this is the essence of ranking. The Learning task uses a form of ranking metric which has been borrowed from the web retrieval search into E-Discovery (Oard and Webber, 2013).

As previously explained, a seed set is given, the seed set is just a list of very relevant documents for a topic. These relevant documents are arrived at after an iterative exploration and analysis by human experts. The goal is to make a ML algorithm to infer patterns of relevance for a topic from this seed set. After training the algorithm, the algorithm assigns a probability of likelihood of relevance to each document in the collection. The higher the probability of a document, the more relevant/responsive it is. Ideally, we would like to see the number of relevant documents among the ranked documents within a particular cutoff point, say, 10,000. The precision, recall, and the F_1 at this cutoff is them computed. Usually, there could be several cutoff points (e.g, k = 5000, 10000,50000,...) which shows the *depth* at which we would like to assess the performance of the retrieval system. The ranking quality can be assessed by observing the F_1 score. The highest F_1 score at each cutoff point is referred to as the *Hypothetical* F_1 and it sets an upper bound on the achievable F_1 score of an actual production (Oard and Webber, 2013).

Topic	Responsive	Non Responsive	Total
401	1040	1460	2500
401	238	1864	2102
403	245	1954	2199

TABLE 6.3: Topic Authority Relevance Determination (Seed Set)

We evaluated our model on the Learning task of the TREC legal track 2011. The seed set was provided as a kind of a *Topic Authority* (TA) relevance determination from the mop-up task. In practice, the participants would perform an initial review, select some documents which are deemed relevant to a topic, and then liaise with the topic authority who then determines whether those documents are relevant. The statistics of the relevance judgment is displayed in table 6.3. In theory, this is a seed set and it contains relevant documents for each topic. Please see (Grossman et al., 2011) for details about how the assessment for responsiveness was carried out. The 2011 task uses the same dataset with the 2010 task. As displayed in table 6.3, participants were given three topics, i.e., 401, 402, and 403. We reproduce the topics below:

1. *Topic 401*.

   > All documents or communications that describe, discuss, refer to, report on, or relate to the design, development, operation, or marketing of enrononline, or any other online service offered, provided, or used by the Company (or any of its subsidiaries, predecessors, or successors-in-interest), for the purchase, sale, trading, or exchange of financial or other instruments or products, including but not limited to, derivative instruments, commodities, futures, and swaps.

2. *Topic 402*.

   > All documents or communications that describe, discuss, refer to, report on, or relate to whether the purchase, sale, trading, or exchange of over-the-counter derivatives, or any other actual or contemplated financial instruments or products, is, was, would be, or will be legal or illegal, or permitted or prohibited, under any existing or proposed rule(s), regulation(s), law(s), standard(s), or other proscription(s), whether domestic or foreign.

3. *Topic 403*.

   > All documents or communications that describe, discuss, refer to, report on, or relate to the environmental impact of any activity or activities undertaken by the Company, including but not limited to, any measures taken to conform to, comply with, avoid, circumvent, or influence any existing or proposed rule(s), regulation(s), law(s), standard(s), or other

proscription(s), such as those governing environmental emissions, spills, pollution, noise, and/or animal habitats

6.12 Discussion

The results of our evaluation for topics 401, 402, and 403 are as shown in the figures 6.6, 6.7, and 6.8 respectively. The evaluation was done using 6 cutoff depths (c), i.e, 2k, 5k, 20k, 50k, 100k, and 200k. We observe a monotonous relationship regarding the depth for both the precision and the recall. While the recall score grows with the depth, the precision does not seem to increase as the depth increases. In particular, this is noticeable in our evaluation for the three topics. For topic 401, we can see that the F_1 score of our system increases drastically with depth initially but falls back at later stages. For instance, between c = 2000 and c = 5k, our model achieves an increase of about 68% improvement, however, by the cutoff point c=200k, the F_1 measure has degraded to 20. Conversely, the best F_1 score was achieved at c=20k. For topic 402, the best F_1 score was achieved at c=5k, while for topic 403, the best F_1 score was achieved at c=5k. Generally, our model significantly outperforms the baseline systems as shown in the result tables. This assessment is based on the hypothetical F_1 scores at each cutoff point. The baseline scores that are shown in the table are the result of the participants who submitted the result of their systems for evaluation at the TREC legal track 2011 by the organizers. The task overview paper (Grossman et al., 2011) is silent about the description of the individual systems used in our comparison.

In the experiment on TREC 2010 Interactive task, we see that we generally obtained better precision scores than the recall. This is indeed for a text classification task where there are two classes. Moreover, even though we lack many information and guidance which are normally provided to participants by the organizers in order to aid the development of their system, we see that our model clearly outperformed the benchmark systems. In the privilege task which is essential to document production, we see an improvement in the recall. An essential characteristic of our system is that it is trained end-to-end to classify and rank documents. This means that we employ the same model for the interactive task as well as the learning task. For instance, for every query-document pair, the system produces two probability scores which are assigned to the Relevant or Not-Relevant classes. Furthermore, these scores determine whether the document is assigned the Relevant class or otherwise. It is possible to rank documents using the learned relevance scores, especially since we are only interested in ranking the relevant documents only.

One of the most recent work on E-Discovery experimented with an unsupervised classifier approach (Ayetiran, 2017). The author introduced three techniques, i.e., a stem-based search which is more or less a keyword matching; a topic-based search which uses the LDA algorithm in modeling the topical structure of documents and query terms and then finds the similarity between the topic vectors;and an approach which combines the two methods. The author also introduced a disambiguation technique for performing query expansion. Figures 6.9, 6.10, and 6.11 show the comparison of our model to the system proposed by the author. Specifically, we represent the combined approach by the identifier ENI-COMB-UNSUP, the topic-based approach by ENI-TOPIC, and the stem-based approach by ENI-STEM. The comparison is made regarding the TREC 2011 Learning task. We observe that the author's topic-based approach seems to give the best performance, however, our model outperformed this system significantly. In fairness, the system obtains a higher recall score at 2000 cut-off for topics 401 and 402. However, the performance degrades with the depth of cut-off. The initial gain of the topic-based system is understandable for it is tied to lexical matching which ordinarily would degrade once more documents are examined. A system that fails to scale with data may not rightly function in a real-life scenarios. Our approach shows steady improvement in both recall and precision as the depth increases. This implies that it can be employed in real life scenario where even more data are to be reviewed. The main advantage of our system is the incorporation of many relevance signals, while a neural network components identifies positions of matching between the texts, another looks for semantic relatedness (STRF), yet another neural network (LTI) is learning to discover the local and global term intersection through the hierarchical interaction between document and the query texts. In a way, the system benefits from the combination of different strategies.

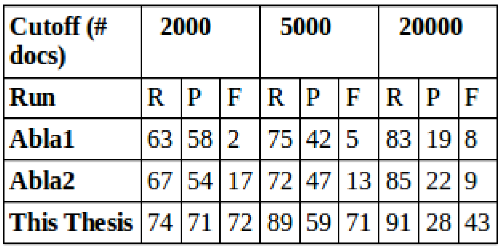
Comparing results in an E-Discovery task depends not only on the techniques proposed but also how the query is formulated. For instance, where this is done manually by an expert, it is possible to obtain an improved performance compared to when the formulation is automatically done. Most of the benchmarked systems have access to human query formulation processes since the teams are allowed to perform the task manually or automatically. In our work, this process has been automatically done by relying on our query expansion method which incorporates explicit knowledge from many sources.

6.12.1 Ablation Experiment

In order to determine the significance of the ensemble model, we performed an ablation experiment where we removed some components neural network (for some features) and then retrain the model. The goal is to see the importance of the components that we removed. Figure 6.5 shows the result of the ablation work. *Abla1* is the result obtained when we removed the LTI and STRF components of our Ensemble model, while *Abla2* shows the result obtained when only the LTI is removed. The reader can notice a degrade in performance when both LTI and STRF are removed at the same time. This is particularly the same when only LTI was removed. Furthermore, the performance degrades seriously after the 20000 cut-off, hence our reporting of the result for just the 2000, 3000, and 5000 cut-offs. In general, we notice the significant improvement when all the features are incorporated.

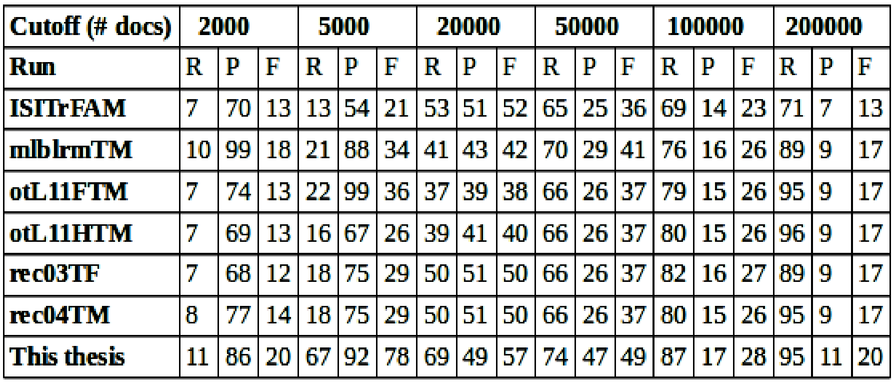
Topic	Team	Recall	Precision	F_1
301	CS	.165	.579	.256
	IT	.205	.295	.242
	SF	.239	.193	.214
	IS	.027	.867	.052
	UW	.019	.578	.036
	This Thesis	**.391**	**.881**	**.541**
302	UM	.200	.450	.277
	UW	.169	.732	.275
	MM	.115	.410	.180
	LA	.096	.481	.160
	IS	.090	.693	.160
	IN	.135	.017	.031
	This Thesis	**.295**	**.820**	**.433**
303	EQ	.801	.577	.671
	CB2	.572	.705	.631
	CB1	.452	.734	.559
	UB	.723	.300	.424
	IT	.248	.259	.254
	UW	.134	.773	.228
	This Thesis	**.815**	**.770**	**.791**
304	CB3	.633	.302	.408
	CB4	.715	.264	.385
	CB2	.271	.402	.324
	CB1	.201	.327	.249
	IN	.072	.494	.126
	This Thesis	**.736**	**.425**	**.538**

TABLE 6.4: Comparison With TREC Legal Track 2010 Interactive Task Submitted Systems



Cutoff (# docs)	2000			5000			20000		
Run	R	P	F	R	P	F	R	P	F
Abla1	63	58	2	75	42	5	83	19	8
Abla2	67	54	17	72	47	13	85	22	9
This Thesis	74	71	72	89	59	71	91	28	43

FIGURE 6.5: Ablation Result on Topic 402 Recall (%), Precision (%), and F_1 at representative document review cutoffs for Legal 2011 Learning Task

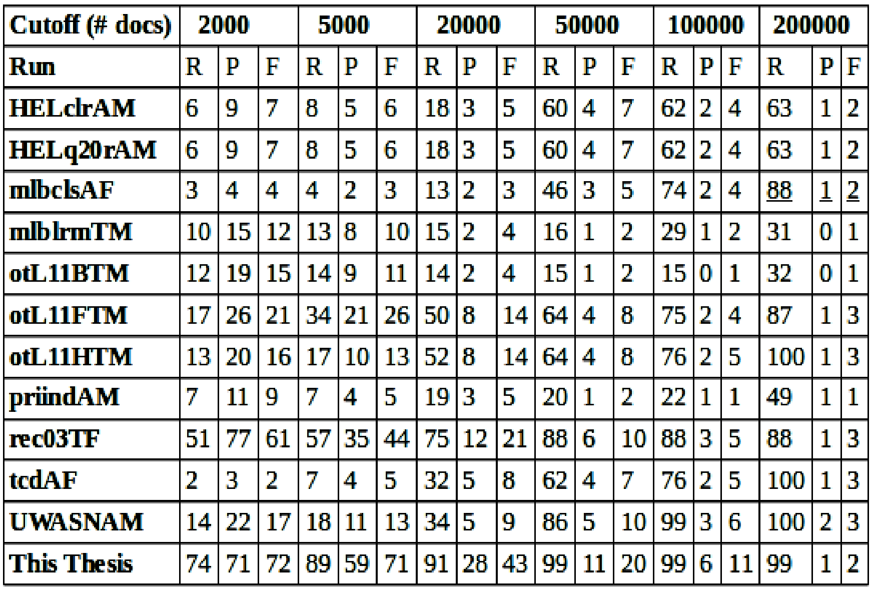


Cutoff (# docs)	2000			5000			20000			50000			100000			200000		
Run	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F
ISITrFAM	7	70	13	13	54	21	53	51	52	65	25	36	69	14	23	71	7	13
mlblrmTM	10	99	18	21	88	34	41	43	42	70	29	41	76	16	26	89	9	17
otL11FTM	7	74	13	22	99	36	37	39	38	66	26	37	79	15	26	95	9	17
otL11HTM	7	69	13	16	67	26	39	41	40	66	26	37	80	15	26	96	9	17
rec03TF	7	68	12	18	75	29	50	51	50	66	26	37	82	16	27	89	9	17
rec04TM	8	77	14	18	75	29	50	51	50	66	26	37	80	15	26	95	9	17
This thesis	11	86	20	67	92	78	69	49	57	74	47	49	87	17	28	95	11	20

FIGURE 6.6: Topic 401 Recall (%), Precision (%), and F_1 at representative document review cutoffs for Legal 2011 Learning Task

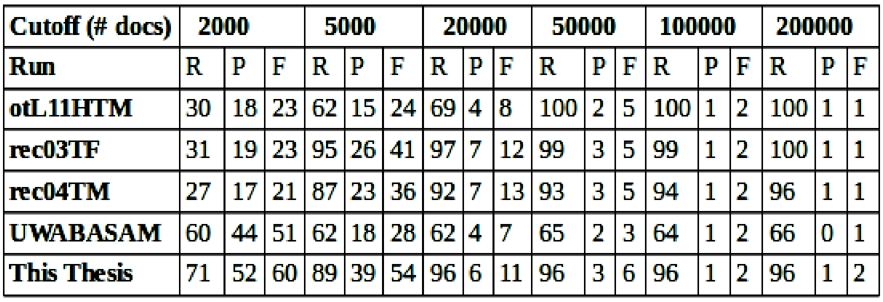
6.13 Chapter Summary

In this chapter, we described a Neural Network-based classifier which has been developed in the context of E-Discovery search. As already discussed, E-Discovery incorporates ideas from the general IR task while also adding some distinctive features. For instance, the main task is that of classifying whether a document is relevant or not. This is more or less a text classification task. We have shown that a Neural Network classifier is appropriate for this task. Our model being an ensemble system incorporates many relevance features. In particular, the model performs a form of *feature fusion*, i.e., combining knowledge from traditional IR approaches in order to ensure a good relevance matching. The results from our evaluation justifies our methodology. Even though a few work already utilized Neural Network for information retrieval, this has been restricted mostly to *Web Search* using *Click-through* data, e.g., see (DSSM -Huang et al., 2013; DUET -Mitra, Diaz, and Craswell, 2017; DESM -Mitra et al., 2016; C-DSSM -Shen et al., 2014; and MatchPyramid -Pang et al., 2016). These studies performed evaluation using Web data whose distinguishing feature is divergent to that of E-Discovery. Researchers in Legal Information research have hitherto focused on *SVM* for text classification in E-Discovery. A factor that can greatly affect the performance of Machine Learning classifiers is the quality of relevance judgment. However, that is beyond the scope of this study. Obviously, an error-prone data would lead to an unimaginable level of randomness in prediction. Researchers have also studied this and reached an empirical conclusion (Voorhees,



Cutoff (# docs)	2000			5000			20000			50000			100000			200000		
Run	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F
HELclrAM	6	9	7	8	5	6	18	3	5	60	4	7	62	2	4	63	1	2
HELq20rAM	6	9	7	8	5	6	18	3	5	60	4	7	62	2	4	63	1	2
mlbclsAF	3	4	4	4	2	3	13	2	3	46	3	5	74	2	4	88	1	2
mlblrmTM	10	15	12	13	8	10	15	2	4	16	1	2	29	1	2	31	0	1
otL11BTM	12	19	15	14	9	11	14	2	4	15	1	2	15	0	1	32	0	1
otL11FTM	17	26	21	34	21	26	50	8	14	64	4	8	75	2	4	87	1	3
otL11HTM	13	20	16	17	10	13	52	8	14	64	4	8	76	2	5	100	1	3
priindAM	7	11	9	7	4	5	19	3	5	20	1	2	22	1	1	49	1	1
rec03TF	51	77	61	57	35	44	75	12	21	88	6	10	88	3	5	88	1	3
tcdAF	2	3	2	7	4	5	32	5	8	62	4	7	76	2	5	100	1	3
UWASNAM	14	22	17	18	11	13	34	5	9	86	5	10	99	3	6	100	2	3
This Thesis	74	71	72	89	59	71	91	28	43	99	11	20	99	6	11	99	1	2

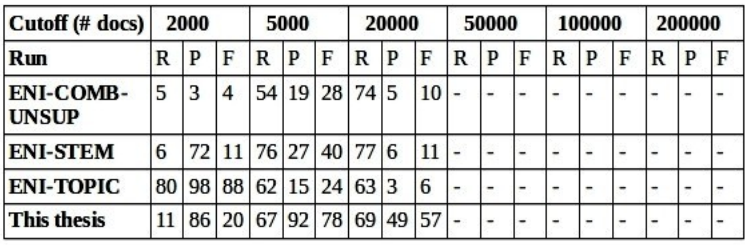
FIGURE 6.7: Topic 402 Recall (%), Precision (%), and F_1 at representative document review cutoffs for Legal 2011 Learning Task



Cutoff (# docs)	2000			5000			20000			50000			100000			200000		
Run	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F
otL11HTM	30	18	23	62	15	24	69	4	8	100	2	5	100	1	2	100	1	1
rec03TF	31	19	23	95	26	41	97	7	12	99	3	5	99	1	2	100	1	1
rec04TM	27	17	21	87	23	36	92	7	13	93	3	5	94	1	2	96	1	1
UWABASAM	60	44	51	62	18	28	62	4	7	65	2	3	64	1	2	66	0	1
This Thesis	71	52	60	89	39	54	96	6	11	96	3	6	96	1	2	96	1	2

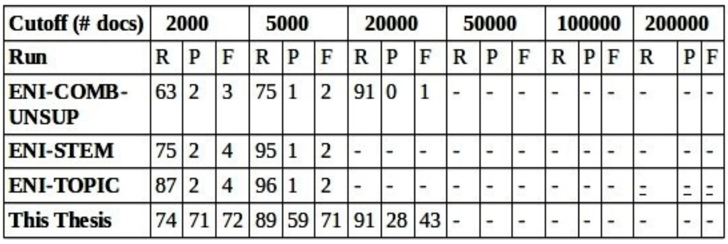
FIGURE 6.8: Topic 403 Recall (%), Precision (%), and F_1 at representative document review cutoffs for Legal 2011 Learning Task.

2000; Wang and Soergel, 2010; Webber and Pickens, 2013). This study, at least to the best of our knowledge represents the first adaptation of Deep Learning techniques to the E-Discovery problem. More importantly, the proposed approach is the first that combines different approaches to relevance which have been modeled separately by individual Neural Network components and then combined with another Neural Network. We have empirically demonstrated that the performance of the system is convincing when evaluated on the TREC Legal track 2010 Interactive task and the 2011 Learning task.



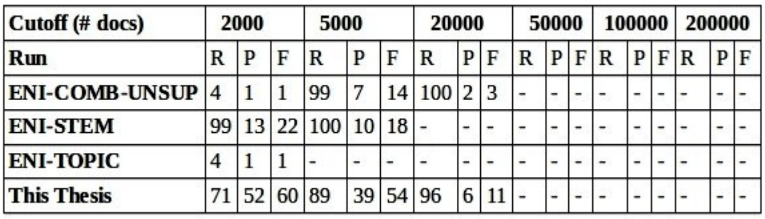
Cutoff (# docs)	2000			5000			20000			50000			100000			200000		
Run	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F
ENI-COMB-UNSUP	5	3	4	54	19	28	74	5	10	-	-	-	-	-	-	-	-	-
ENI-STEM	6	72	11	76	27	40	77	6	11	-	-	-	-	-	-	-	-	-
ENI-TOPIC	80	98	88	62	15	24	63	3	6	-	-	-	-	-	-	-	-	-
This thesis	11	86	20	67	92	78	69	49	57	-	-	-	-	-	-	-	-	-

FIGURE 6.9: Comparative analysis of performance with a set of unsupervised techniques on Topic 401.
Recall (%), Precision (%), and F_1 at representative document review cutoffs for Legal 2011 Learning Task



Cutoff (# docs)	2000			5000			20000			50000			100000			200000		
Run	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F
ENI-COMB-UNSUP	63	2	3	75	1	2	91	0	1	-	-	-	-	-	-	-	-	-
ENI-STEM	75	2	4	95	1	2	-	-	-	-	-	-	-	-	-	-	-	-
ENI-TOPIC	87	2	4	96	1	2	-	-	-	-	-	-	-	-	-	-	-	-
This Thesis	74	71	72	89	59	71	91	28	43	-	-	-	-	-	-	-	-	-

FIGURE 6.10: Comparative analysis of performance with a set of unsupervised techniques on Topic 402.
Recall (%), Precision (%), and F_1 at representative document review cutoffs for Legal 2011 Learning Task



Cutoff (# docs)	2000			5000			20000			50000			100000			200000		
Run	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F
ENI-COMB-UNSUP	4	1	1	99	7	14	100	2	3	-	-	-	-	-	-	-	-	-
ENI-STEM	99	13	22	100	10	18	-	-	-	-	-	-	-	-	-	-	-	-
ENI-TOPIC	4	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
This Thesis	71	52	60	89	39	54	96	6	11	-	-	-	-	-	-	-	-	-

FIGURE 6.11: Comparative analysis of performance with a set of unsupervised techniques on Topic 403.
Recall (%), Precision (%), and F_1 at representative document review cutoffs for Legal 2011 Learning Task.

Part IV

Conclusion and Future Work

Chapter 7

Conclusion And Future Work

The Information Retrieval field and the Legal domain are seemingly routinely becoming like a Siamese-twin. Lawyers and legal practitioners are more than ever inundated with a massive amount of information to handle and they have learned to rely on the expertise of Information retrieval researchers. There are several information needs of Legal Practitioners. A common example that is of huge economic importance is E-Discovery. In the United States of America, about 19,303,000 civil cases are filed in state and federal courts respectively each year, with about 60% of involving discovery. Discovery alone cost the United States an average of $*42.1* billion per year. Experts have estimated the cost in the range of $200 - $250 billion. In particular, the E-Discovery software business is estimated to eclipse $16 billion by the year 2021. On top of this is the need to also manage court documents, parliamentary documents, and other documents that lawyers have to deal with day-to-day. It is obvious that the Legal Information research requires a systemic modeling and conceptualization of task-specific requirement and developing custom solutions to cater for each problem.

This research provides a parallel distillation of the task-specific needs of legal experts. We view the general Legal Information retrieval as a diverse set of tasks, each requiring a custom built solution. We then analyze each problem and propose an adequate solution.

In this thesis, we have developed solutions which benefit from the state of the art techniques in natural language processing. First, we developed a conceptual retrieval system, relying on a fusion of some natural language processing techniques like topical text segmentation, explicit semantic analysis, text similarity, and semantic annotation. The evaluation approach is to estimate how accurately the model maps a legal concept to a text snippet. This forms a crucial basis for the conceptual information retrieval which works at the level of text semantics. The system simplifies document retrieval, e.g., a legal text collection like the EurLex can be conceptually queried using either a vocabulary or non-vocabulary controlled concept. Our evaluation shows that the technique is not only novel but performed creditably. In particular, the conceptual analysis module of the semantic annotator was redeployed for query expansion in our experiment on E-Discovery.

The final part of our research shows our Neural Network model for ad-hoc search. E-Discovery is essentially an ad-hoc search which can have a tremendous social, political and economic impact on the society. We describe our Neural Network model which focuses both on semantic matching and relevance matching. As a matter of fact, traditional approaches mainly focus on linguistic/lexical matching and it is a common knowledge that they fail grossly where synonyms and polysemous words are at play. Semantic matching, on the other hand, focuses on realizing the meaning of the query and the meaning of the text, and conversely, mapping query to text based on their meaning. However, while this might be sufficient for some types of information retrieval, it is not sufficient in a recall-oriented search like the E-Discovery. Coupled with the huge cost of missing out on any relevant document, no organization or lawyer would take a risk of relying solely on semantic matching. We discovered that even though lexical matching is an old and empirically unstable method, it still works well in some cases. We identified some important features that we can derive from the use of lexical and semantic matching, and combined them appropriately to perform what we called relevance matching. In a sense, separate neural network components look for different relevance signals, including semantic relatedness. We found out that relatedness is particularly important for a large-scale search like the E-Discovery. Our model therefore encodes knowledge which is induced from training a large document collection. This readily captures semantic relatedness and similarity between terms. Word embedding models which are trained on a large corpora are readily useful for IR task in the legal domain.

The basic of our work is a Neural Network which learned to match a relevant document to a query while teaching itself to identify non-relevant ones for the same query. We evaluated our technique on the Learning task of TREC Legal Track 2011 and the Interactive task of TREC Legal Track 2010. The evaluation shows a strong performance across boards, while significantly outperforming the result submitted by participants during those years.

Chapter 8

Resources, tools, and links to their sources

8.1 Datasets

Dataset	Chapter	Source
EUR-Lex	5	http://www.ke.tu-darmstadt.de/resources/eurlex/
Wikipedia	5	https://dumps.wikimedia.org/
TREC	6	https://trec-legal.umiacs.umd.edu/
JRC	5	https://ec.europa.eu/jrc/en/language-technologies/jrc-acquis
Choi	5	http://www.cs.man.ac.uk/~mary/choif/software.html

8.2 Software tools

Software	Source
Gensim	https://github.com/RaRe-Technologies/gensim
Keras	https://github.com/keras-team/keras
Faiss	https://github.com/facebookresearch/faiss

8.3 Other Resources

Resources	Source
Eurovoc	http://eurovoc.europa.eu/
Eurlex	https://eur-lex.europa.eu/homepage.html
WordNet/NLTK	https://www.nltk.org/
GloVe	http://nlp.stanford.edu/data/glove.840B.300d.zip

Bibliography

Adebayo, John Kolawole, Luigi Di Caro, and Guido Boella (2017a). “Siamese Network with Soft Attention for Semantic Text Understanding”. In: *Semantics 2017 Association for Computing Machinery (ACM)*. ACM.

– (2017b). *Solving Bar Exams with Deep Neural Network*. Vol. -. Easychair.

Adebayo, John Kolawole et al. (2017). “Legalbot: A Deep Learning-Based Conversational Agent in the Legal Domain”. In: *International Conference on Applications of Natural Language to Information Systems*. Springer, pp. 267–273.

Adebayo, Kolawole J et al. (2016a). “Textual Inference with Tree-structured LSTMs”. In: pp. 17–31.

Adebayo, Kolawole John, Luigi Di Caro, and Guido Boella (2016a). “A Supervised KeyPhrase Extraction System”. In: *Proceedings of the 12th International Conference on Semantic Systems*. ACM, pp. 57–62.

– (2016b). “Neural Reasoning for Legal Text Understanding”. In: *Legal Knowledge and Information Systems - JURIX 2016: The Twenty-Ninth Annual Conference*, pp. 175–178. DOI: 10.3233/978-1-61499-726-9-175. URL: https://doi.org/10.3233/978-1-61499-726-9-175.

– (2016c). “Normas at semeval-2016 task 1: Semsim: A multi-feature approach to semantic text similarity”. In: *Proceedings of SemEval*, pp. 718–725.

– (2016d). “NORMAS at SemEval-2016 Task 1: SEMSIM: A Multi-Feature Approach to Semantic Text Similarity”. In: *Proceedings of SemEval*, pp. 718–725.

– (2016e). “Text Segmentation with Topic Modeling and Entity Coherence”. In: *International Conference on Hybrid Intelligent Systems*. Springer, pp. 175–185.

– (2017c). “Semantic annotation of legal document with ontology concepts”. In: *AICOL 2015 LNCS PROCEEDINGS*. Springer.

Adebayo, Kolawole John et al. (2016b). “An approach to information retrieval and question answering in the legal domain”. In: pp. 15–25.

Ai, Qingyao et al. (2016). “Improving language estimation with the paragraph vector model for ad-hoc retrieval”. In: *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, pp. 869–872.

Ajani, Gianmaria et al. (2007). “Terminological and ontological analysis of european directives: multilinguism in law”. In: *Proceedings of the 11th international conference on Artificial intelligence and law*. ACM, pp. 43–48.

Almquist, Brian Alan (2011). *Mining for evidence in enterprise corpora*. The University of Iowa.

Arazy, Ofer (2004). *Artificial semantics in text retrieval*. University of British Columbia.

Arguello, Jaime et al. (2008). "Document Representation and Query Expansion Models for Blog Recommendation." In: *ICWSM* 2008.0, p. 1.

Auer, Soren et al. (2007). "Dbpedia: A nucleus for a web of open data". In: *The semantic web*, pp. 722–735.

Auttonberry, L Casey (2013). "Predictive Coding: Taking the Devil Out of the Details". In: *La. L. Rev.* 74, p. 613.

Ayetiran, Eniafe Festus (2017). "A Combined Unsupervised Technique for Automatic Classification in Electronic Discovery". PhD thesis. University of Bologna, Italy.

Baeza-Yates, Ricardo, Berthier Ribeiro-Neto, et al. (1999). *Modern information retrieval*. Vol. 463. ACM press New York.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014). "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473*.

Bakx, Gerard Escudero, LM Villodre, and GR Claramunt (2006). "Machine learning techniques for word sense disambiguation". In: *Unpublished doctoral dissertation, Universitat Politecnica de Catalunya*.

Banerjee, Satanjeev and Ted Pedersen (2002). "An adapted Lesk algorithm for word sense disambiguation using WordNet". In: *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, pp. 136–145.

Baron, Jason R (2011). "Law in the Age of Exabytes: Some Further Thoughts on'Information Inflation'and Current Issues in E-Discovery Search". In: *Rich. JL & Tech.* 17, pp. 9–16.

Baron, Jason R et al. (2007). "The sedona conference best practices commentary on the use of search and information retrieval methods in e-discovery". In: *The Sedona conference journal*. Vol. 8, pp. 189–223.

Baroni, Marco (2013). "Composition in distributional semantics". In: *Language and Linguistics Compass* 7.10, pp. 511–522.

Baroni, Marco, Raffaela Bernardi, and Roberto Zamparelli (2014). "Frege in space: A program of compositional distributional semantics". In: *LiLT (Linguistic Issues in Language Technology)* 9.

Barzilay, Regina and Mirella Lapata (2008). "Modeling local coherence: An entity-based approach". In: *Computational Linguistics* 34.1, pp. 1–34.

Beeferman, Doug, Adam Berger, and John Lafferty (1999). "Statistical models for text segmentation". In: *Machine learning* 34.1-3, pp. 177–210.

Bench-Capon, Trevor et al. (2012). "A history of AI and Law in 50 papers: 25 years of the international conference on AI and Law". In: *Artificial Intelligence and Law* 20.3, pp. 215–319.

Bengio, Yoshua et al. (2003). "A neural probabilistic language model". In: *Journal of machine learning research* 3.Feb, pp. 1137–1155.

Bengio, Yoshua et al. (2009). "Learning deep architectures for AI". In: *Foundations and trends in Machine Learning* 2.1, pp. 1–127.

Berners-Lee, Tim, James Hendler, Ora Lassila, et al. (2001). "The semantic web". In: *Scientific american* 284.5, pp. 28–37.

Bhogal, Jagdev, Andrew MacFarlane, and Peter Smith (2007). "A review of ontology based query expansion". In: *Information processing and management* 43.4, pp. 866–886.

Bikakis, Nikos et al. (2010). "Integrating keywords and semantics on document annotation and search". In: *On the Move to Meaningful Internet Systems, OTM 2010*. Springer, pp. 921–938.

Bishop, Christopher M (2006). *Pattern recognition and machine learning*. springer.

Blair, David C and Melvin E Maron (1985). "An evaluation of retrieval effectiveness for a full-text document-retrieval system". In: *Communications of the ACM* 28.3, pp. 289–299.

Blei, David M and John D Lafferty (2006). "Dynamic topic models". In: *Proceedings of the 23rd international conference on Machine learning*. ACM, pp. 113–120.

Blei, David M, Andrew Y Ng, and Michael I Jordan (2003). "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan, pp. 993–1022.

Boella, Guido et al. (2012a). "Eunomos, a legal document and knowledge management system for regulatory compliance". In: *Information systems: crossroads for organization, management, accounting and engineering*. Springer, pp. 571–578.

Boella, Guido et al. (2012b). "NLP challenges for eunomos, a tool to build and manage legal knowledge". In: *Language Resources and Evaluation (LREC)*, pp. 3672–3678.

Boella, Guido et al. (2016). "Eunomos, a legal document and knowledge management system for the web to provide relevant, reliable and up-to-date information on the law". In: *Artificial Intelligence and Law* 24.3, pp. 245–283.

Bordes, Antoine and Jason Weston (2016). "Learning end-to-end goal-oriented dialog". In: *arXiv preprint arXiv:1605.07683*.

Bordes, Antoine et al. (2015). "Large-scale simple question answering with memory networks". In: *arXiv preprint arXiv:1506.02075*.

Borlund, Pia (2003). "The concept of relevance in IR". In: *Journal of the Association for Information Science and Technology* 54.10, pp. 913–925.

Brill, Eric and Robert C Moore (2000). "An improved error model for noisy channel spelling correction". In: *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 286–293.

Buckley, Chris, James Allan, and Gerard Salton (1994). "Automatic routing and ad-hoc retrieval using SMART: TREC 2". In: *NIST SPECIAL PUBLICATION SP*, pp. 45–45.

Buckley, Chris et al. (1995). "Automatic query expansion using SMART: TREC 3". In: *NIST special publication sp*, pp. 69–69.

Burges, Chris et al. (2005). "Learning to rank using gradient descent". In: *Proceedings of the 22nd international conference on Machine learning*. ACM, pp. 89–96.

Burges, Christopher JC (2010). "From ranknet to lambdarank to lambdamart: An overview". In: *Learning* 11.23-581, p. 81.

Cabrio, Elena et al. (2012). "Qakis@ qald-2". In: *2nd open challenge in Question Answering over Linked Data (QALD-2)*.

Cabrio, Elena et al. (2013). "Querying multilingual dbpedia with qakis". In: *Extended Semantic Web Conference*. Springer, pp. 194–198.

Callan, James P (1994). "Passage-level evidence in document retrieval". In: *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*. Springer-Verlag New York, Inc., pp. 302–310.

Callan, James P, W Bruce Croft, and Stephen M Harding (1992). "The INQUERY retrieval system". In: *Proceedings of the third international conference on database and expert systems applications*, pp. 78–83.

Carpineto, Claudio and Giovanni Romano (2012). "A survey of automatic query expansion in information retrieval". In: *ACM Computing Surveys (CSUR)* 44.1, p. 1.

Chang, Chih-Chung and Chih-Jen Lin (2011). "LIBSVM: a library for support vector machines". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 2.3, p. 27.

Charton, Eric and Michel Gagnon (2012). "A disambiguation resource extracted from Wikipedia for semantic annotation." In: *LREC*, pp. 3665–3671.

Chen, Chungmin Melvin and Nick Roussopoulos (1994). *Adaptive selectivity estimation using query feedback*. Vol. 23. 2. ACM.

Cho, Kyunghyun et al. (2014). "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078*.

Choi, Freddy YY (2000). "Advances in domain independent linear text segmentation". In: *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*. Association for Computational Linguistics, pp. 26–33.

Choi, Freddy YY, Peter Wiemer-Hastings, and Johanna Moore (2001). "Latent semantic analysis for text segmentation". In: *In Proceedings of EMNLP*. Citeseer.

Clinchant, Stéphane and Florent Perronnin (2013). "Aggregating continuous word embeddings for information retrieval". In: *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pp. 100–109.

Cormack, Gordon V and Maura R Grossman (2014). "Evaluation of machine-learning protocols for technology-assisted review in electronic discovery". In: *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM, pp. 153–162.

Cormack, Gordon V et al. (2010). "Overview of the TREC 2010 legal track". In: *Proc. 19th Text REtrieval Conference*, p. 1.

Cortes, Corinna and Vladimir Vapnik (1995). "Support-vector networks". In: *Machine learning* 20.3, pp. 273–297.

Crestani, Fabio (1994). "Comparing neural and probabilistic relevance feedback in an interactive information retrieval system". In: *Neural Networks, 1994. IEEE World Congress on Computational Intelligence., 1994 IEEE International Conference on*. Vol. 5. IEEE, pp. 3426–3430.

Croft, Bruce and John Lafferty (2013). *Language modeling for information retrieval*. Vol. 13. Springer Science & Business Media.

Croft, W Bruce, Donald Metzler, and Trevor Strohman (2010). *Search engines: Information retrieval in practice*. Vol. 283. Addison-Wesley Reading.

Cunningham, Hamish et al. (2002). "A framework and graphical development environment for robust NLP tools and applications." In: *ACL*, pp. 168–175.

Daelemans, Walter and Katharina Morik (2008). *Machine Learning and Knowledge Discovery in Databases: European Conference, Antwerp, Belgium, September 15-19, 2008, Proceedings*. Vol. 5212. Springer.

Dahl, George E et al. (2012). "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition". In: *IEEE Transactions on audio, speech, and language processing* 20.1, pp. 30–42.

Dai, Andrew M, Christopher Olah, and Quoc V Le (2015). "Document embedding with paragraph vectors". In: *arXiv preprint arXiv:1507.07998*.

Deerwester, Scott et al. (1990). "Indexing by latent semantic analysis". In: *Journal of the American society for information science* 41.6, p. 391.

Dias, Gael, Elsa Alves, and Jose Gabriel Pereira Lopes (2007). "Topic segmentation algorithms for text summarization and passage retrieval: An exhaustive evaluation". In: *AAAI*. Vol. 7, pp. 1334–1339.

Dill, Stephen et al. (2003a). "A case for automated large-scale semantic annotation". In: *Web Semantics: Science, Services and Agents on the World Wide Web* 1.1, pp. 115–132.

Dill, Stephen et al. (2003b). "SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation". In: *Proceedings of the 12th international conference on World Wide Web*. ACM, pp. 178–186.

Du, Lan, John K Pate, and Mark Johnson (2015). "Topic segmentation in an ordering-based topic model". In:

Duan, Huizhong and Bo-June Paul Hsu (2011). "Online spelling correction for query completion". In: *Proceedings of the 20th international conference on World wide web*. ACM, pp. 117–126.

Egozi, Ofer, Shaul Markovitch, and Evgeniy Gabrilovich (2011). "Concept-based information retrieval using explicit semantic analysis". In: *ACM Transactions on Information Systems (TOIS)* 29.2, p. 8.

Eisenstein, Jacob (2009). "Hierarchical text segmentation from multi-scale lexical cohesion". In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 353–361.

Fader, Anthony, Stephen Soderland, and Oren Etzioni (2011). "Identifying relations for open information extraction". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 1535–1545.

Fader, Anthony, Luke Zettlemoyer, and Oren Etzioni (2014). "Open question answering over curated and extracted knowledge bases". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 1156–1165.

Fernandez-Barrera, Meritxell and Pompeu Casanovas (2011). "Towards the intelligent processing of non-expert generated content: Mapping web 2.0 data with ontologies in the domain of consumer mediation". In: *Proceedings of the ICAIL Workshop, Applying Human Language Technology to the Law*, pp. 18–27.

Firth, John R (1957). "A synopsis of linguistic theory, 1930-1955". In:

Gabrilovich, Evgeniy and Shaul Markovitch (2006). "Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge". In: *AAAI*. Vol. 6, pp. 1301–1306.

– (2007). "Computing semantic relatedness using wikipedia-based explicit semantic analysis." In: *IJcAI*. Vol. 7, pp. 1606–1611.

Gangemi, Aldo et al. (2002). "Sweetening ontologies with DOLCE". In: *Knowledge engineering and knowledge management: Ontologies and the semantic Web*, pp. 223–233.

Gao, Jianfeng, Kristina Toutanova, and Wen-tau Yih (2011). "Clickthrough-based latent semantic models for web search". In: *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, pp. 675–684.

Goodfellow, Ian et al. (2014). "Generative adversarial nets". In: *Advances in neural information processing systems*, pp. 2672–2680.

Greengrass, Ed (2000). "Information retrieval: A survey". In:

Grefenstette, Edward (2013). "Towards a formal distributional semantics: Simulating logical calculi with tensors". In: *arXiv preprint arXiv:1304.5823*.

Grefenstette, Edward et al. (2014). "Concrete sentence spaces for compositional distributional models of meaning". In: *Computing Meaning*. Springer, pp. 71–86.

Grossman, Maura R and Gordon V Cormack (2010). "Technology-assisted review in e-discovery can be more effective and more efficient than exhaustive manual review". In: *Rich. JL & Tech.* 17, p. 1.

Grossman, Maura R et al. (2011). "Overview of the TREC 2011 Legal Track." In: *TREC*. Vol. 11.

Grosz, Barbara J, Scott Weinstein, and Aravind K Joshi (1995). "Centering: A framework for modeling the local coherence of discourse". In: *Computational linguistics* 21.2, pp. 203–225.

Gruber, Thomas R (1993). "A translation approach to portable ontology specifications". In: *Knowledge acquisition* 5.2, pp. 199–220.

Guo, Jiafeng et al. (2008). "A unified and discriminative model for query refinement". In: *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 379–386.

Guo, Jiafeng et al. (2016). "A deep relevance matching model for ad-hoc retrieval". In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, pp. 55–64.

Halliday, Michael Alexander Kirkwood and Ruqaiya Hasan (2014). *Cohesion in english*. Routledge.

Hampton, Wallis (2014). *Predictive Coding: It's Here to Stay*. Vol. -. practicallaw.com Thomson Reuters.

Handschuh, Siegfried and Steffen Staab (2002). "Authoring and annotation of web pages in CREAM". In: *Proceedings of the 11th international conference on World Wide Web*. ACM, pp. 462–473.

Harris, Zellig S (1954). "Distributional structure". In: *Word* 10.2-3, pp. 146–162.

Harter, Stephen P (1992). "Psychological relevance and information science". In: *Journal of the American Society for information Science* 43.9, p. 602.

Hartigan, John A and Manchek A Wong (1979). "Algorithm AS 136: A k-means clustering algorithm". In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28.1, pp. 100–108.

He, Kaiming et al. (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Hearst, Marti A (1993). *TextTiling: A quantitative approach to discourse segmentation*. Tech. rep. Citeseer.

– (1994). "Multi-paragraph segmentation of expository text". In: *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 9–16.

– (1997). "TextTiling: Segmenting text into multi-paragraph subtopic passages". In: *Computational linguistics* 23.1, pp. 33–64.

Hedin, Bruce et al. (2009). *Overview of the TREC 2009 legal track*. Tech. rep. NATIONAL ARCHIVES and RECORDS ADMINISTRATION COLLEGE PARK MD.

Hiemstra, Djoerd (1998). "A linguistically motivated probabilistic model of information retrieval". In: *Research and advanced technology for digital libraries*, pp. 515–515.

– (2009). "Information retrieval models". In: *Information Retrieval: searching in the 21st Century*, pp. 2–19.

Hinton, Geoffrey et al. (2012). "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups". In: *IEEE Signal Processing Magazine* 29.6, pp. 82–97.

Hochreiter, S. and J. Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8.

Hofmann, Thomas (1999). "Probabilistic latent semantic indexing". In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 50–57.

Hou, Jun (2014). "Text mining with semantic annotation: using enriched text representation for entity-oriented retrieval, semantic relation identification and text clustering". PhD thesis. Queensland University of Technology.

Hu, Baotian et al. (2014). "Convolutional neural network architectures for matching natural language sentences". In: *Advances in neural information processing systems*, pp. 2042–2050.

Huang, Po-Sen et al. (2013). "Learning deep structured semantic models for web search using clickthrough data". In: *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM, pp. 2333–2338.

Hyman, Harvey (2012). *Learning and Relevance in Information Retrieval: A Study in the Application of Exploration and User Knowledge to Enhance Performance*. University of South Florida.

Joachims, Thorsten (1998). "Text categorization with support vector machines: Learning with many relevant features". In: *Machine learning: ECML-98*, pp. 137–142.

Joulin, Armand et al. (2016). “Bag of Tricks for Efficient Text Classification”. In: *arXiv preprint arXiv:1607.01759*.

Kaszkiel, Marcin and Justin Zobel (1997). “Passage retrieval revisited”. In: *ACM SIGIR Forum*. Vol. 31. SI. ACM, pp. 178–185.

Kaufmann, Stefan (1999). “Cohesion and collocation: Using context vectors in text segmentation”. In: *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*. Association for Computational Linguistics, pp. 591–595.

Kemp, DA (1974). “Relevance, pertinence and information system development”. In: *Information Storage and Retrieval* 10.2, pp. 37–47.

Kiyavitskaya, Nadzeya et al. (2005). “Semi-Automatic Semantic Annotations for Web Documents.” In: *SWAP*.

Kiyavitskaya, Nadzeya et al. (2006). “Text mining through semi automatic semantic annotation”. In: *International Conference on Practical Aspects of Knowledge Management*. Springer, pp. 143–154.

Korfhage, Robert R (2008). “Information storage and retrieval”. In:

Kumar, Ankit et al. (2016). “Ask me anything: Dynamic memory networks for natural language processing”. In: *International Conference on Machine Learning*, pp. 1378–1387.

Laclavik, Michal et al. (2006). “Ontology based text annotation–OnTeA”. In: *Proc. of*, pp. 280–284.

Laclavik, Michal et al. (2007). “Ontea: Semi-automatic pattern based text annotation empowered with information retrieval methods”. In: *Tools for acquisition, organisation and presenting of information and knowledge: Proceedings in Informatics and Information Technologies, Kosice, Vydavatelstvo STU, Bratislava, part* 2, pp. 119–129.

Lancaster, Frederick Wilfrid and Emily Gallup (1973). “Information retrieval on-line”. In:

Landauer, Thomas K, Peter W Foltz, and Darrell Laham (1998). “An introduction to latent semantic analysis”. In: *Discourse processes* 25.2-3, pp. 259–284.

Lavrenko, Victor and W Bruce Croft (2001). “Relevance based language models”. In: *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 120–127.

Le, Quoc and Tomas Mikolov (2014). “Distributed representations of sentences and documents”. In: *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1188–1196.

Leckie, Gloria J, Karen E Pettigrew, and Christian Sylvain (1996). “Modeling the information seeking of professionals: A general model derived from research on engineers, health care professionals, and lawyers”. In: *The Library Quarterly* 66.2, pp. 161–193.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). “Deep learning”. In: *Nature* 521.7553, pp. 436–444.

Lee, Dik L, Huei Chuang, and Kent Seamons (1997). “Document ranking and the vector-space model”. In: *IEEE software* 14.2, pp. 67–75.

Lee, Joon Ho (1995). "Combining multiple evidence from different properties of weighting schemes". In: *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 180–188.

Li, Hang (2011). "A short introduction to learning to rank". In: *IEICE TRANSACTIONS on Information and Systems* 94.10, pp. 1854–1862.

– (2014). "Learning to rank for information retrieval and natural language processing". In: *Synthesis Lectures on Human Language Technologies* 7.3, pp. 1–121.

Li, Hang, Jun Xu, et al. (2014). "Semantic matching in search". In: *Foundations and Trends in Information Retrieval* 7.5, pp. 343–469.

Li, Yinghao et al. (2007). "Improving weak ad-hoc queries using wikipedia as external corpus". In: *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 797–798.

Li, Yuhua et al. (2006). "Sentence similarity based on semantic nets and corpus statistics". In: *Knowledge and Data Engineering, IEEE Transactions on* 18.8, pp. 1138–1150.

Liaw, Andy, Matthew Wiener, et al. (2002). "Classification and regression by randomForest". In: *R news* 2.3, pp. 18–22.

Liu, Tie-Yan et al. (2009). "Learning to rank for information retrieval". In: *Foundations and Trends® in Information Retrieval* 3.3, pp. 225–331.

Liu, Xiaoyong and W Bruce Croft (2005). *Statistical language modeling for information retrieval*. Tech. rep. MASSACHUSETTS UNIV AMHERST CENTER FOR INTELLIGENT INFORMATION RETRIEVAL.

Luhn, Hans Peter (1957). "A statistical approach to mechanized encoding and searching of literary information". In: *IBM Journal of research and development* 1.4, pp. 309–317.

Lyytikainen, VIRPI, PASI Tiitinen, and AIRI Salminen (2000). "Challenges for European legal information retrieval". In: *Proceedings of the IFIP 8.5 Working Conference on Advances in Electronic Government*, pp. 121–132.

Mann, William C and Sandra A Thompson (1988). "Rhetorical structure theory: Toward a functional theory of text organization". In: *Text-Interdisciplinary Journal for the Study of Discourse* 8.3, pp. 243–281.

Manning, Christopher D, Prabhakar Raghavan, Hinrich Schutze, et al. (2008). *Introduction to information retrieval*. Vol. 1. 1. Cambridge university press Cambridge.

Manning, Christopher D et al. (2014). "The Stanford CoreNLP Natural Language Processing Toolkit." In: *ACL (System Demonstrations)*, pp. 55–60.

Maron, Melvin Earl and John L Kuhns (1960). "On relevance, probabilistic indexing and information retrieval". In: *Journal of the ACM (JACM)* 7.3, pp. 216–244.

Mencia, Eneldo Loza and Johannes Furnkranz (2010). "Efficient multilabel classification algorithms for large-scale problems in the legal domain". In: *Semantic Processing of Legal Texts*. Springer, pp. 192–215.

Metzler, Donald (2011). *A feature-centric view of information retrieval*. Vol. 27. Springer Science & Business Media.

Michalski, Ryszard S, Jaime G Carbonell, and Tom M Mitchell (2013). *Machine learning: An artificial intelligence approach*. Springer Science & Business Media.

Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig (2013). “Linguistic regularities in continuous space word representations.” In: *hlt-Naacl*. Vol. 13, pp. 746–751.

Mikolov, Tomas et al. (2013a). “Distributed representations of words and phrases and their compositionality”. In: *Advances in neural information processing systems*, pp. 3111–3119.

Mikolov, Tomas et al. (2013b). “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781*.

Miller, George A (1995). “WordNet: a lexical database for English”. In: *Communications of the ACM* 38.11, pp. 39–41.

Misra, Hemant et al. (2009). “Text segmentation via topic modeling: an analytical study”. In: *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, pp. 1553–1556.

Misra, Hemant et al. (2011). “Text segmentation: A topic modeling perspective”. In: *Information Processing and Management* 47.4, pp. 528–544.

Mitchell, Patrick C (1974). “A note about the proximity operators in information retrieval”. In: *ACM SIGIR Forum*. Vol. 9. 3. ACM, pp. 177–180.

Mitra, Bhaskar and Nick Craswell (2017a). “Neural Models for Information Retrieval”. In: *arXiv preprint arXiv:1705.01509*.

– (2017b). “Neural Text Embeddings for Information Retrieval”. In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, pp. 813–814.

Mitra, Bhaskar, Fernando Diaz, and Nick Craswell (2017). “Learning to Match using Local and Distributed Representations of Text for Web Search”. In: *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, pp. 1291–1299.

Mitra, Bhaskar et al. (2016). “A dual embedding space model for document ranking”. In: *arXiv preprint arXiv:1602.01137*.

Moens, Marie-Francine (2001). “Innovative techniques for legal text retrieval”. In: *Artificial Intelligence and Law* 9.1, pp. 29–57.

Mommers, Laurens (2010). “Ontologies in the legal domain”. In: *Theory and Applications of Ontology: Philosophical Perspectives*. Springer, pp. 265–276.

Monique, Altheim (2011). *Edicovery, European Union Data Protection, Online Privacy*. Vol. 4. ediscoverymap.com.

Navigli, Roberto (2009). “Word sense disambiguation: A survey”. In: *ACM Computing Surveys (CSUR)* 41.2, p. 10.

Newman, Mark EJ (2005). “Power laws, Pareto distributions and Zipf’s law”. In: *Contemporary physics* 46.5, pp. 323–351.

Noortwijk, Kees van, Johanna Visser, and Richard V De Mulder (2006). “Ranking and classifying legal documents using conceptual information”. In: *The Journal of Information Law and Technology* 2006.1.

Oard, Douglas W, William Webber, et al. (2013). “Information retrieval for e-discovery”. In: *Foundations and Trends in Information Retrieval* 7.2–3, pp. 99–237.

Oard, Douglas W et al. (2008). *Overview of the TREC 2008 legal track*. Tech. rep. MARYLAND UNIV COLLEGE PARK COLL OF INFORMATION STUDIES.

Oard, Douglas W et al. (2010). "Evaluation of information retrieval for E-discovery". In: *Artificial Intelligence and Law* 18.4, pp. 347–386.

Palangi, Hamid et al. (2016). "Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval". In: *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)* 24.4, pp. 694–707.

Palmirani, Monica and Fabio Vitali (2011). "Akoma-Ntoso for legal documents". In: *Legislative XML for the semantic Web*. Springer, pp. 75–100.

– (2012). *Legislative XML: principles and technical tools*. Tech. rep. Inter-American Development Bank.

Pang, Liang et al. (2016). "A study of matchpyramid models on ad-hoc retrieval". In: *arXiv preprint arXiv:1606.04648*.

Parikh, Ankur P et al. (2016). "A decomposable attention model for natural language inference". In: *arXiv preprint arXiv:1606.01933*.

Park, Taemin Kim (1993). "The nature of relevance in information retrieval: An empirical study". In: *The library quarterly* 63.3, pp. 318–351.

Passonneau, Rebecca J and Diane J Litman (1997). "Discourse segmentation by human and automated means". In: *Computational Linguistics* 23.1, pp. 103–139.

Pennington, Jeffrey, Richard Socher, and Christopher D Manning (2014). "Glove: Global Vectors for Word Representation." In: *EMNLP*. Vol. 14, pp. 1532–43.

Pevzner, Lev and Marti A Hearst (2002). "A critique and improvement of an evaluation metric for text segmentation". In: *Computational Linguistics* 28.1, pp. 19–36.

Pohl, Stefan (2012). *Boolean and Ranked Information Retrieval for Biomedical Systematic Reviewing*. University of Melbourne, Department of Computer Science and Software Engineering.

Ponte, Jay M and W Bruce Croft (1998). "A language modeling approach to information retrieval". In: *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 275–281.

Popov, Borislav et al. (2003). "KIM–semantic annotation platform". In: *The Semantic Web-ISWC 2003*. Springer, pp. 834–849.

Pouliquen, Bruno, Ralf Steinberger, and Camelia Ignat (2006). "Automatic annotation of multilingual text collections with a conceptual thesaurus". In: *arXiv preprint cs/0609059*.

Presutti, Valentina, Francesco Draicchio, and Aldo Gangemi (2012). "Knowledge extraction based on discourse representation theory and linguistic frames". In: *International Conference on Knowledge Engineering and Knowledge Management*. Springer, pp. 114–129.

Reeve Jr, Lawrence Harold (2006). "Semantic Annotation and Summarization of Biomedical Literature". PhD thesis. Drexel University.

Remus, Dana A (2013). "The Uncertain Promise of Predictive Coding". In: *Iowa L. Rev.* 99, p. 1691.

Resnik, Philip (1995). "Using information content to evaluate semantic similarity in a taxonomy". In: *arXiv preprint cmp-lg/9511007*.

Reynar, Jeffrey C (1999). “Statistical models for topic segmentation”. In: *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*. Association for Computational Linguistics, pp. 357–364.

Riedl, Martin and Chris Biemann (2012a). “Text segmentation with topic models”. In: *Journal for Language Technology and Computational Linguistics* 27.1, pp. 47–69.

– (2012b). “TopicTiling: A Text Segmentation Algorithm based on LDA”. In: *Proceedings of ACL 2012 Student Research Workshop*. Association for Computational Linguistics, pp. 37–42.

Robertson, Stephen, Hugo Zaragoza, and Michael Taylor (2004). “Simple BM25 extension to multiple weighted fields”. In: *Proceedings of the thirteenth ACM international conference on Information and knowledge management*. ACM, pp. 42–49.

Robertson, Stephen, Hugo Zaragoza, et al. (2009). “The probabilistic relevance framework: BM25 and beyond”. In: *Foundations and Trends® in Information Retrieval* 3.4, pp. 333–389.

Robertson, Stephen E (1977). “The probability ranking principle in IR”. In: *Journal of documentation* 33.4, pp. 294–304.

– (1990). “On term selection for query expansion”. In: *Journal of documentation* 46.4, pp. 359–364.

Robertson, Stephen E and K Sparck Jones (1976). “Relevance weighting of search terms”. In: *Journal of the Association for Information Science and Technology* 27.3, pp. 129–146.

Robertson, Stephen E, Cornelis J van Rijsbergen, and Martin F Porter (1980). “Probabilistic models of indexing and searching”. In: *Proceedings of the 3rd annual ACM conference on Research and development in information retrieval*. Butterworth & Co., pp. 35–56.

Robertson, Stephen E and Steve Walker (1994). “Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval”. In: *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*. Springer-Verlag New York, Inc., pp. 232–241.

Robertson, Stephen E et al. (1995). “Okapi at TREC-3”. In: *Nist Special Publication Sp* 109, p. 109.

Rocchio, Joseph John (1971). “Relevance feedback in information retrieval”. In: *The Smart retrieval system-experiments in automatic document processing*.

Rohan, Nanda et al. (2017). “Legal Information Retrieval Using Topic Clustering and Neural Networks”. In: *COLIEE 2017. 4th Competition on Legal Information Extraction and Entailment, held in conjunction with the 16th International Conference on Artificial Intelligence and Law (ICAIL 2017) in King's College London, UK*. Pp. 68–78. URL: http://www.easychair.org/publications/paper/347228.

Roitblat, Herbert L, Anne Kershaw, and Patrick Oot (2010). “Document categorization in legal electronic discovery: computer classification vs. manual review”. In: *Journal of the Association for Information Science and Technology* 61.1, pp. 70–80.

Rosenblatt, Frank (1958). “The perceptron: A probabilistic model for information storage and organization in the brain.” In: *Psychological review* 65.6, p. 386.

Rosso, Paolo, Santiago Correa, and Davide Buscaldi (2011). "Passage retrieval in legal texts". In: *The Journal of Logic and Algebraic Programming* 80.3-5, pp. 139–153.

Sahlgren, Magnus (2008). "The distributional hypothesis". In: *Italian Journal of Disability Studies* 20, pp. 33–53.

Salton, Gerard (1968). "Automatic information organization and retrieval". In:

– (1971). "The SMART retrieval system—experiments in automatic document processing". In:

Salton, Gerard, James Allan, and Chris Buckley (1993). "Approaches to passage retrieval in full text information systems". In: *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 49–58.

Salton, Gerard and Chris Buckley (1997). "Improving retrieval performance by relevance feedback". In: *Readings in information retrieval* 24.5, pp. 355–363.

Salton, Gerard and Christopher Buckley (1988). "Term-weighting approaches in automatic text retrieval". In: *Information processing & management* 24.5, pp. 513–523.

Salton, Gerard, Edward A Fox, and Harry Wu (1983). "Extended Boolean information retrieval". In: *Communications of the ACM* 26.11, pp. 1022–1036.

Salton, Gerard and Michael J McGill (1986). "Introduction to modern information retrieval". In:

Salton, Gerard, Anita Wong, and Chung-Shu Yang (1975). "A vector space model for automatic indexing". In: *Communications of the ACM* 18.11, pp. 613–620.

Saracevic, Tefko (1975). "Relevance: A review of and a framework for the thinking on the notion in information science". In: *Journal of the Association for Information Science and Technology* 26.6, pp. 321–343.

– (1996). "Relevance reconsidered". In: *Proceedings of the second conference on conceptions of library and information science (CoLIS 2)*, pp. 201–218.

Saric, Frane et al. (2012). "Takelab: Systems for measuring semantic text similarity". In: *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. Association for Computational Linguistics, pp. 441–448.

Schamber, Linda, Michael B Eisenberg, and Michael S Nilan (1990). "A re-examination of relevance: toward a dynamic, situational definition". In: *Information processing and management* 26.6, pp. 755–776.

Schmidhuber, Jurgen (2015). "Deep learning in neural networks: An overview". In: *Neural networks* 61, pp. 85–117.

Schweighofer, Erich, Anton Geist, et al. (2007). "Legal Query Expansion using Ontologies and Relevance Feedback." In: *LOAIT*, pp. 149–160.

Schweighofer, Erich and Doris Liebwald (2007). "Advanced lexical ontologies and hybrid knowledge based systems: First steps to a dynamic legal electronic commentary". In: *Artificial Intelligence and Law* 15.2, pp. 103–115.

Sebastiani, Fabrizio (2002). "Machine learning in automated text categorization". In: *ACM computing surveys (CSUR)* 34.1, pp. 1–47.

Severyn, A and A Mochitti (2015). "Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks". In: *SIGIR*, pp. 373–382.

Shen, Yelong et al. (2014). "Learning semantic representations using convolutional neural networks for web search". In: *Proceedings of the 23rd International Conference on World Wide Web*. ACM, pp. 373–374.

Simonyan, Karen and Andrew Zisserman (2014). "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556*.

Singhal, Amit (2001). "Modern information retrieval: A brief overview". In: *IEEE Data Eng. Bull.* 24.4, pp. 35–43.

Singhal, Amit, Chris Buckley, and Mandar Mitra (1996). "Pivoted document length normalization". In: *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 21–29.

Socha, GJ and T Gelbmann (2005). "The Electronic Discovery Reference Model Project (EDRM)". In:

Song, Fei and W Bruce Croft (1999). "A general language model for information retrieval". In: *Proceedings of the eighth international conference on Information and knowledge management*. ACM, pp. 316–321.

Sormunen, Eero (2001). "Extensions to the STAIRS study—empirical evidence for the hypothesised ineffectiveness of Boolean queries in large full-text databases". In: *Information Retrieval* 4.3, pp. 257–273.

Sutskever, Ilya, Oriol Vinyals, and Quoc V Le (2014). "Sequence to sequence learning with neural networks". In: *Advances in neural information processing systems*, pp. 3104–3112.

Szegedy, Christian et al. (2013). "Intriguing properties of neural networks". In: *arXiv preprint arXiv:1312.6199*.

Tegos, Athanasios, Vangelis Karkaletsis, and Alexandros Potamianos (2008). "Learning of semantic relations between ontology concepts using statistical techniques". In: *High-level Information Extraction Workshop*.

Tellex, Stefanie et al. (2003). "Quantitative evaluation of passage retrieval algorithms for question answering". In: *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. ACM, pp. 41–47.

Turney, Peter D, Patrick Pantel, et al. (2010). "From frequency to meaning: Vector space models of semantics". In: *Journal of artificial intelligence research* 37.1, pp. 141–188.

Usbeck, Ricardo et al. (2015). "GERBIL: general entity annotator benchmarking framework". In: *Proceedings of the 24th International Conference on World Wide Web*. ACM, pp. 1133–1143.

Utiyama, Masao and Hitoshi Isahara (2001). "A statistical model for domain-independent text segmentation". In: *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 499–506.

Uzuner, Ozlem, Boris Katz, Deniz Yuret, et al. (1999). "Word sense disambiguation for information retrieval". In: *AAAI* 985.

Van Opijnen, Marc and Cristiana Santos (2017). "On the concept of relevance in legal information retrieval". In: *Artificial Intelligence and Law* 25.1, pp. 65–87.

Voorhees, Ellen M (1994). “Query expansion using lexical-semantic relations”. In: *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*. Springer-Verlag New York, Inc., pp. 61–69.

– (2000). “Variations in relevance judgments and the measurement of retrieval effectiveness”. In: *Information processing & management* 36.5, pp. 697–716.

– (2001). “The philosophy of information retrieval evaluation”. In: *CLEF*. Vol. 1. Springer, pp. 355–370.

Voorhees, Ellen M, Donna K Harman, et al. (2005). *TREC: Experiment and evaluation in information retrieval*. Vol. 1. MIT press Cambridge.

Wang, Jianqiang and Dagobert Soergel (2010). “A user study of relevance judgments for E-Discovery”. In: *Proceedings of the Association for Information Science and Technology* 47.1, pp. 1–10.

Wang, Zhiguo, Wael Hamza, and Radu Florian (2017). “Bilateral Multi-Perspective Matching for Natural Language Sentences”. In: *arXiv preprint arXiv:1702.03814*.

Webber, William and Jeremy Pickens (2013). “Assessor disagreement and text classifier accuracy”. In: *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 929–932.

Wei, Xing (2007). *Topic models in information retrieval*. University of Massachusetts Amherst.

Wilkinson, Ross (1994). “Effective retrieval of structured documents”. In: *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*. Springer-Verlag New York, Inc., pp. 311–317.

Wilson, Patrick (1973). “Situational relevance”. In: *Information storage and retrieval* 9.8, pp. 457–471.

Witten, Ian H and David N Milne (2008). “An effective, low-cost measure of semantic relatedness obtained from Wikipedia links”. In:

Xu, Jinxi and W Bruce Croft (1996). “Query expansion using local and global document analysis”. In: *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 4–11.

Yarowsky, David (1995). “Unsupervised word sense disambiguation rivaling supervised methods”. In: *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 189–196.

Yoon, Kim (2014). “Convolutional neural networks for sentence classification”. In: *arXiv preprint arXiv:1408.5882*.

Yu, Lei et al. (2014). “Deep learning for answer sentence selection”. In: *arXiv preprint arXiv:1412.1632*.

Zavitsanos, Elias et al. (2010). “Scalable Semantic Annotation of Text Using Lexical and Web Resources.” In: *SETN*. Springer, pp. 287–296.

Zeleznikow, John et al. (2005). “Knowledge discovery from legal databases”. In:

Zhai, Chengxiang and John Lafferty (2001). “Model-based feedback in the language modeling approach to information retrieval”. In: *Proceedings of the tenth international conference on Information and knowledge management*. ACM, pp. 403–410.

Zobel, Justin and Alistair Moffat (1998). “Exploring the similarity space”. In: *ACM SIGIR Forum*. Vol. 32. 1. ACM, pp. 18–34.

Zurada, Jacek M (1992). *Introduction to artificial neural systems*. Vol. 8. West St. Paul.