Developing Unsupervised Knowledge-Enhanced Models to Reduce the Semantic Gap in Information Retrieval

Stefano Marchesin

Supervisor: Prof. Maristella Agosti

Department of Information Engineering
University of Padua

This dissertation is submitted for the degree of
Doctor of Philosophy

September 2020

To my family.

Acknowledgements

First, I would like to thank my supervisor, Prof. Maristella Agosti, without whom this dissertation would have not been possible. Maristella guided me throughout these years, always leaving me the freedom to choose the things I believed in. Most importantly, she taught me how to be a researcher.

Secondly, I would also like to thank my reviewers, Prof. Giuseppe Amato from the Consiglio Nazionale delle Ricerche ISTI – Istituto di Scienza e Tecnologia dell'Informazione, and Prof. Francesco Ciompi from the Computational Pathology Group, Department of Pathology, Radboud University Medical Center. Their insightful comments improved this dissertation in many ways.

Then, I would like to acknowledge the senior members of the Information Management Systems (IMS) Research Group, Prof. Nicola Ferro, Prof. Giorgio Maria Di Nunzio, and Prof. Gianmaria Silvello. I am grateful to have had the opportunity to work with each of them, as well as for their constant support during all these years.

I would also like to acknowledge all my past and present colleagues at the IMS group. In particular, Dr. Maria Maistro, Federica Vezzani, Dennis Dosso, and Alberto Purpura. We've been through a lot of things together, and I am lucky to have been able to share this journey with them.

A special thanks to my friends, who lighten my life and make difficult days less difficult.

Last but not least, I would not be where I am today without the incredible support, encouragement, and love of my parents, Fiorella and Vladimiro, and my brother, Alessandro. All this would not have been possible without them, and I dedicate this milestone to them. Finally, I would like to thank Gaia, who has been by my side throughout all this journey, living every single moment of it, and without whom, it would not have been the wonderful journey it has been. This paragraph is not enough to express my gratitude for the support and love I received from all of them.

This work was partially supported by the ExaMode project, as part of the European Union H2020 program under Grant Agreement no. 825292.

Abstract

In this thesis we tackle the semantic gap, a long-standing problem in Information Retrieval (IR). The semantic gap can be described as the mismatch between users' queries and the way retrieval models answer to such queries. Two main lines of work have emerged over the years to bridge the semantic gap: (i) the use of external knowledge resources to enhance the bag-of-words representations used by lexical models, and (ii) the use of semantic models to perform matching between the latent representations of queries and documents. To deal with this issue, we first perform an in-depth evaluation of lexical and semantic models through different analyses. The objective of this evaluation is to understand what features lexical and semantic models share, if their signals are complementary, and how they can be combined to effectively address the semantic gap. In particular, the evaluation focuses on (semantic) neural models and their critical aspects. Then, we build on the insights of this evaluation to develop lexical and semantic models addressing the semantic gap. Specifically, we develop unsupervised models that integrate knowledge from external resources, and we evaluate them for the medical domain – a domain with a high social value, where the semantic gap is prominent, and the large presence of authoritative knowledge resources allows us to explore effective ways to leverage external knowledge to address the semantic gap. For lexical models, we propose and evaluate several knowledge-based query expansion and reduction techniques. These query reformulations are used to increase the probability of retrieving relevant documents by adding to or removing from the original query highly specific terms. Regarding semantic models, we first analyze the limitations of the knowledge-enhanced neural models presented in the literature. Then, to overcome these limitations, we propose SAFIR, an unsupervised knowledge-enhanced neural framework for IR. The representations learned within this framework are optimized for IR and encode linguistic features that are relevant to address the semantic gap.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Information has always played a leading role in human history. For thousand of years, people have perceived the importance of storing, maintaining, and retrieving information. The need to store and retrieve written information became increasingly important with the invention of paper, printing press, and eventually computers. With the advent of computers, people realized that they could exploit them to store and access large amounts of information [35]. Meanwhile, the explosion of scientific publications that emerged during and after World War II – and the need to efficiently and effectively search this literature – motivated the development of automatic systems capable of searching through a large collection of documents, known as *corpus*, to find and retrieve relevant information addressing a particular information need, typically expressed by a user through a keyword *query* [19]. Starting from the 1950s, several works proposed to search text using computers. In particular, Luhn [151] proposed to use words as indexing units for documents and to estimate word overlap as a criterion for retrieval. Modern Information Retrieval (IR) was born, and with it, the first automatic IR systems began to take hold.

An IR system takes as input a query – formulated by a user to express their information need – and returns a ranking list of documents potentially relevant to that query. Although these systems were introduced in the 1950s [197], several key developments in the field of IR happened in the 1960s. Among them, the development of the System for the Mechanical Analysis and Retrieval of Text (SMART) [195] by Salton and his group, along with the Cranfield evaluations carried out by Cleverdon and his colleagues [43, 44], allowed the IR field to advance rapidly. Building on the pioneering works performed in 1960s, several models for document retrieval were developed in 1970s and 1980s, pushing research forward in all dimensions of the retrieval process. However, back then, the proposed models and techniques were experimentally evaluated on small text collections, consisting of several thousand articles only. Therefore, the ability of retrieval models to scale to large collections remained an open question. In 1992, the first Text REtrieval Conference (TREC) changed this situation [100]. Sponsored by the National Institute of Standards and Technology (NIST), TREC aimed to build large text collections and foster IR research under a common evaluation framework. Over the years, TREC has evolved into a series of evaluation campaigns that provide the infrastructure necessary for large-scale evaluation of IR systems. Thus, since the first TREC, many different models and techniques have been developed – and are still being developed – to perform efficient and effective retrieval over large collections.

However, the advent of large collections highlighted the limitations of traditional retrieval models, which compute the relevance score using heuristics defined over the lexical overlap between query and document bag-of-words representations. For instance, traditional models – also known as lexical models – fall short when the user's query and a relevant document describe the same concept but with different words (i.e., synonymy mismatch), or when the query and an irrelevant document describe different concepts using the same words (i.e., polysemy mismatch). Put simply, they fail to represent the semantics of queries and documents.

Such limitations are strongly related to a long-standing problem in IR: the semantic gap. The semantic gap can be defined as the mismatch between users' queries and the way IR systems answer to such queries [85, 51]. Depending on the situation, the semantic gap can hinder the retrieval of relevant documents, affect the quality of the produced ranking list, or both. Over the years, two main lines of work have emerged to overcome the limitations of lexical models and address the semantic gap: (i) the use of external knowledge resources to enhance the representations used by lexical models, and (ii) the adoption of semantic models to perform semantic matching between query and document latent representations.

Enhancing lexical models through external knowledge resources dates back to the early years of IR [194]. In literature, the relational information contained within knowledge resources has been used to model linguistic features related to the semantic gap, like synonymy and polysemy, and enhance the bag-of-words representations used by lexical models. However, even though several approaches have been proposed, many open questions remain. For example, what knowledge resources are best suited to enhance bag-of-words representations? What information stored within knowledge resources help to address the semantic gap? To what extent external knowledge can be integrated within bag-of-words representations without incurring into noise injection?

On the other hand, semantic models have been used for decades in IR as a means to bridge the semantic gap and retrieve relevant documents that lexical models fail to discover [140]. Semantic models perform semantic matching between queries and documents in a latent semantic space. Compared to lexical matching, semantic matching computes the similarity between two elements – be them a pair of words, or a query/document pair – as the distance, under a given metric, between their low-dimensional latent representations. In this way, semantic models overcome the limitations related to (lexical) exact matching and capture the similarity between queries and documents at a higher semantic level. Nevertheless, traditional semantic models [57, 113, 28] have never been up to par with lexical models due to their coarseness and lack of specificity [68, 238]. In the last few years, the advances and the success of neural representation learning in several different tasks have promoted the diffusion of neural models in IR, reviving the interest in semantic matching. In particular, neural models based on the distributional hypothesis [102] have shown promising results when used to address the semantic gap between queries and documents. However, these models suffer from two main limitations: they fail to discriminate polysemous words, as the different meanings of a word are conflated into a single representation; and they fail to learn close representations for synonyms occurring in different contexts, as they lack the relational knowledge required to identify synonymy relationships between words. To overcome these limitations, recent works that integrate external knowledge into the learning process of neural models have been proposed in the Natural Language Processing (NLP) community, but only a few have been applied in IR to reduce the effect of the semantic gap between queries and documents [147, 169, 168, 170].

The work of this thesis falls within both the lines of research presented above and aims to advance research on lexical and semantic models to reduce the semantic gap between queries and documents. In particular, we are interested in models that can be applied without the need for labeled data and can be used at the early stages of the IR pipeline. Developing models not requiring labeled data allows us to address the semantic gap in any domain – and in particular in those domains with a high social value, where labeled data are scarce and expensive resources (e.g., medicine). On the other hand, developing models that can be used at the early stages of the IR pipeline is fundamental to address the semantic gap. Otherwise, relevant documents most affected by the semantic gap will simply remain undiscovered. Hence, we focus on the development of unsupervised models that integrate external knowledge to address the semantic gap at first-stage retrieval.

1.1 Objectives and Contributions

Since we aim to address the semantic gap between queries and documents to improve the retrieval performances of lexical and semantic models, the main objectives of this thesis are:

- Study lexical and semantic signals to understand if they are complementary and how they can be combined to effectively address the semantic gap.

- Investigate the use of external resources for the development of knowledge-enhanced lexical models.
- Investigate the potential of knowledge-enhanced semantic models for first-stage retrieval.

To achieve our first objective, we perform an in-depth evaluation of lexical and semantic models through different analyses [158]. Each analysis brings a different perspective in the understanding of semantic models and their relation with lexical models. In particular, the evaluation focuses on the critical aspects of (semantic) neural models.

Based on the insights of this in-depth evaluation, we investigate the use of external knowledge resources to enhance lexical and semantic models and address the semantic gap. We develop unsupervised models that integrate relational knowledge from external resources, and we evaluate them in the medical domain. The medical domain is a domain with a high social value, where the semantic gap is prominent [70, 130, 131], and the large presence of authoritative knowledge resources – manually curated by professionals – enables us to explore effective ways to integrate external knowledge within retrieval models to address the semantic gap.

For lexical models, we propose different methods to exploit knowledge from external resources. In particular, we investigate how, and to what extent, concepts and relations stored within knowledge resources can be integrated in query representations to improve the effectiveness of lexical models. In this regard, we present a series of studies and analyses on the TREC Precision Medicine (PM) Track.[1]

First, we conduct a preliminary study [4] on the TREC PM 2018 Clinical Trials task, where the goal is the retrieval of relevant clinical trials for which a target patient is eligible. In this respect, we propose a method to: 1) expand queries iteratively – relying on medical knowledge resources – to increase the probability of finding relevant trials, and 2) filter out trials for which the target patient is not eligible. The objective of the study is to evaluate how a recall-oriented approach based on increasing – and more aggressive – query expansions affects precision. In particular, we investigate whether the retrieval performance can be correlated with the quality of the relational information contained within the knowledge resource(s) used for the expansion process.

Then, we deepen the analysis and we extend it to both TREC PM tasks – i.e., to scientific literature and clinical trials retrieval [5]. We propose and evaluate several knowledge-based query expansion and reduction techniques to investigate whether a particular approach can be helpful in both scientific literature and clinical trials retrieval. The analysis contributes

[1] http://www.trec-cds.org/

to understanding the effectiveness of query reformulation techniques and sheds light on the different characteristics of the considered tasks.

Given the outcomes of the in-depth analysis, we conduct a validation study on the TREC PM 2019 Track [61]. We focus on both tasks, with a particular emphasis on clinical trials retrieval, and we evaluate how the developed query reformulations affect the results and whether the findings obtained in the previous analysis remain valid. Then, we explore the effectiveness of combining different query reformulations in such a highly specific scenario.

Finally, we perform an a posteriori analysis on the effectiveness of the proposed query reformulations for clinical trials retrieval over the three years of TREC PM [6]. This systematic analysis compares our approach with those proposed by the research groups that participated in all the three years of TREC PM and aims to identify a subset of query reformulations effective on the different sets of topics provided across the years.

Regarding semantic models, we first analyze the knowledge-enhanced neural models employed in the literature to address the semantic gap between queries and documents. The purpose of this study is to understand their critical aspects and evaluate their retrieval performances. The study emphasizes the inability of these knowledge-enhanced neural models to effectively encode relevant features for IR, as well as the need for models capable of providing effective performances at the early stages of the IR pipeline – where the integration of external knowledge can express its full potential.

To overcome the above limitations, we propose the Semantic-Aware Neural Framework for IR (SAFIR) [9], an unsupervised knowledge-enhanced neural framework for IR. SAFIR learns representations that are optimized for IR and encodes linguistic features relevant to address the semantic gap between queries and documents. SAFIR can be applied to any domain where external knowledge resources are available, and it does not require any labeled data for training. We conduct an experimental evaluation to compare SAFIR with other knowledge-enhanced neural models on the TREC Clinical Decision Support (CDS) Track[2] – where the objective is to retrieve relevant medical literature given a medical case report. We consider two retrieval strategies in the experiments: document retrieval and query expansion. Document retrieval gives us the opportunity to investigate the effectiveness of integrating external knowledge into neural models for the typical retrieval scenario, where systems retrieve a set of candidate documents given a query. Query expansion allows us to investigate the effectiveness of knowledge-enhanced neural models – which are specifically designed to address the semantic gap – in providing expansion terms effective at reducing the semantic gap for lexical models. In other words, with query expansion we explore the combination of lexical and semantic models to address the semantic gap at the early stages of the IR pipeline.

[2] http://www.trec-cds.org/

The evaluation contributes to understanding the ability of SAFIR in addressing the semantic gap, as well as the effectiveness of combining lexical and semantic models at the early stages of the IR pipeline.

1.2 Outline

The thesis is organized as follows. In Chapter 2, we describe the two types of resources required to investigate the problem of the semantic gap and achieve the objectives of this thesis – that is, test collections and knowledge resources. On top of that, we also report on the test collections and knowledge resources used in our work. In Chapter 3, we provide the necessary background on the approaches that have been proposed in the literature to address the different aspects of the semantic gap. In Chapter 4, we perform an in-depth evaluation of lexical and semantic models through different analyses. Each analysis brings a different perspective in the understanding of lexical and semantic models. Then, based on the outcomes of the analyses performed in Chapter 4, we investigate the use of external knowledge resources to enhance lexical and semantic models and address the semantic gap. In Chapter 5, we present a series of studies and analyses on knowledge-enhanced query reformulations for precision medicine. The methods developed can be used at the early stages of the IR pipeline to enhance retrieval models and reduce the semantic gap between queries and documents. In Chapter 6, we first analyze the knowledge-enhanced neural models employed in the literature to address the semantic gap. Then, to overcome their limitations, we present a novel unsupervised knowledge-enhanced neural framework that learns representations optimized for IR and encodes linguistic features relevant to address the semantic gap. Finally, in Chapter 7, we report some general conclusions and future directions. As a side note, the main chapters of the thesis present a self-contained structure that helps a reader solely interested in any of them to obtain all the necessary notions required to their understanding.

1.3 Publications

Part of the results of this thesis has been published in relevant journals and conferences of the Information Retrieval field. The contents of the thesis, as well as its insights and outcomes, represent the core of these publications. Below, we present the list of publications ordered by publication date.

- Stefano Marchesin. 2018. *Case-Based Retrieval Using Document-Level Semantic Networks*. Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2018), p. 1451.

- Maristella Agosti, Giorgio Maria Di Nunzio, and Stefano Marchesin. 2018. *The University of Padua IMS Research Group at TREC 2018 Precision Medicine Track*. Proceedings of the 27th Text REtrieval Conference (TREC 2018), pp. 1-10.

- Maristella Agosti, Giorgio Maria Di Nunzio, and Stefano Marchesin. 2019. *An Analysis of Query Reformulation Techniques for Precision Medicine*. Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019), pp. 973-976.

- Stefano Marchesin, Alberto Purpura, and Gianmaria Silvello. 2019. *Focal Elements of Neural Information Retrieval Models. An Outlook through a Reproducibility Study*. Information Processing & Management (IP&M), in press, pp. 1-28.

- Giorgio Maria Di Nunzio, Stefano Marchesin, and Maristella Agosti. 2019. *Exploring how to Combine Query Reformulations for Precision Medicine*. Proceedings of the 28th Text REtrieval Conference (TREC 2019), pp. 1-14.

- Maristella Agosti, Giorgio Maria Di Nunzio, and Stefano Marchesin. 2020. *A Post-Analysis of Query Reformulation Methods for Clinical Trials Retrieval*. Proceedings of the 28th Italian Symposium on Advanced Database Systems (SEBD 2020), pp. 152-159.

- Maristella Agosti, Stefano Marchesin, and Gianmaria Silvello. 2020. *Learning Unsupervised Knowledge-Enhanced Representations to Reduce the Semantic Gap in Information Retrieval*. ACM Transactions on Information Systems (TOIS), 38(4):1-48.

Chapter 2

Resources

In this thesis, we investigate the problem of the semantic gap in IR and how we can employ authoritative and formal knowledge to reduce it. To this end, we require two types of resources: test collections and knowledge resources. Test collections are re-usable and standardized resources that can be used to evaluate IR systems with respect to the system. On the other hand, knowledge resources provide access to authoritative relational information that can be used by IR systems to address the semantic gap. In this chapter, we describe each resource and we report on the test collections and knowledge resources used in our work. In particular, we focus on medical test collections (Section 2.1) and structured, expert-made, medical knowledge resources (Section 2.2).

2.1 Test Collections

A test collection is the most used tool for evaluating the effectiveness of IR systems and consists of a set of topics describing specific information needs, a set of information objects to be searched (e.g., biomedical literature), and relevance judgments indicating which objects are relevant for which topics. Test collections are based on the pioneering work carried out by Cleverdon [43, 44] at Cranfield College of Aeronautics in the 1960s. The objective of Cleverdon's work was to define a formal methodology for evaluating retrieval strategies. To this end, Cleverdon performed a series of experiments to investigate which of several indexing languages was the best [43].[1] The outcome of Cleverdon's experiments, known as the Cranfield paradigm, laid the foundation of IR evaluation and is still considered a standard.

However, the Cranfield paradigm involved manually judging each document in the test collection to determine its relevance to a given topic. Therefore, due to the high cost of

[1] Indexing languages are a subset of natural languages used to represent documents and queries with the goal of improving retrieval.

performing such an operation – both in terms of time and human involvement – it was not suited for building large test collections. To overcome this limitation, Spärck Jones and Van Rijsbergen [124] introduced the concept of the ideal test collection, along with the requirements it should met and the characteristics it should have. In particular, the authors proposed to use a technique called pooling to create a subset of the documents, typically known as the "pool", to judge for a topic. Documents within the pool for a topic are judged for relevance, whereas documents outside the pool are assumed to be irrelevant to that topic. This efficient solution was later implemented by the Text REtrieval Conference (TREC),[2] the first evaluation campaign in IR which has run since 1992 [100]. TREC objective was – and still is – to support research within the IR community by providing the infrastructure necessary for large-scale evaluation of retrieval methodologies. The TREC campaign works as follows: each year TREC provides a test set, composed of a set of documents and a set of topics. Each research group participating in TREC runs their own IR system on the given corpus and returns to TREC the top retrieved documents for each topic. The returned list of documents for a set of topics is called *run*. Then, TREC pools the returned documents, performs the relevance judgments, and presents a ranking with the systems score.

The huge success gained by the TREC initiative encouraged the creation of other evaluation campaigns. The Conference and Labs of the Evaluation Forum (CLEF), which focuses on European languages,[3] the NII Testbeds and Community for Information access Research (NTCIR), which focuses on Asian languages,[4] and the Forum for Information Retrieval Evaluation (FIRE), which specifically focuses on south Asian ones.[5] All these campaigns carry on the effort of providing test collections for a wide variety of domains and tasks – generating and making available several test collections that enable numerous groups from all over the world to participate in the development of next-generation retrieval systems [233]. Therefore, test collections play a fundamental role in providing the basis to measure and compare the effectiveness of different IR systems and techniques.

Below, we present the test collections used in our work.

2.1.1 OHSUMED

The OHSUMED collection [107] consists of $348,566$ references/documents from MEDLINE, the on-line life sciences-biomedicine information database composed of titles, abstracts, and other bibliographic information from most of the published medical journals.[6] OHSUMED

[2] https://trec.nist.gov/

[3] http://www.clef-initiative.eu/

[4] http://research.nii.ac.jp/ntcir/

[5] http://fire.irsi.res.in/fire/

[6] https://www.nlm.nih.gov/bsd/medline.html

was built by Hersh et al. [107] in 1994, after a series of preliminary experiments aimed at evaluating the use of IR systems in medical practice. Inspired by TREC objectives [99], Hersh et al. developed the OHSUMED collection to bring medical, real world-sized test collections to the research community. The collection was also used in the TREC-9 Filtering Track [191]. OHSUMED contains 106 topics, divided into 63 official topics and 43 pre-test topics – that where rejected from official TREC-9 runs because they had too few relevance judgments. Topics include two fields: *title* (patient description) and *description* (information need).

2.1.2 TREC Clinical Decision Support

The TREC Clinical Decision Support (CDS) collections [188, 189, 184] consist of articles from the Open Access Subset of PubMed Central (PMC), an online digital database of freely available full-text biomedical and life sciences journal literature.[7] TREC CDS 2014 (CDS14) and 2015 (CDS15) contain $733,138$ articles, whereas TREC CDS 2016 (CDS16) extends the document set to $1,255,260$ articles. CDS14 and CDS15 contain 30 topics, each, representing medical case narratives created by expert topic developers. The case narratives describe information such as a patient's medical history, current symptoms, tests performed by a physician to diagnose the patient's condition, the eventual diagnosis, and any steps taken by a physician to treat the patient. Topics are provided in two variants: a *description*, a complete account of the patients' visits, including details such as their vital statistics, drug dosages, etc.; a *summary*, a simplified version of the narrative that contains less irrelevant information. CDS16 contains 30 topics, representing Electronic Health Records (EHR) admission notes curated by physicians from the MIMIC-III data. Specifically, the notes are extracted from the History of Present Illness (HPI) section of the note. The HPI describes information such as a patient's chief complaint, medical history, tests performed by a physician to diagnose the patient's condition, possibly the current diagnosis, and any steps taken by a physician to treat the patient. Topics are provided in three variants: the EHR admission *note* (only the HPI section); a more layman-friendly *description*, which removes much of the jargon and replaces clinical abbreviations with their full forms for better readability; a *summary*, a one-or-two sentences summary of the description.

2.1.3 TREC Precision Medicine

There are two target document sets for TREC Precision Medicine (PM) collections [186, 185, 187]: scientific literature and clinical trials.

[7] https://www.ncbi.nlm.nih.gov/pmc/

Scientific Literature

The Scientific Literature document set for TREC PM 2017 (PM17) and 2018 (PM18) consists of a set of $26,759,399$ MEDLINE abstracts, plus two additional sets: (i) $37,007$ abstracts from recent proceedings of the American Society of Clinical Oncology (ASCO),[8] and (ii) $33,018$ abstracts from recent proceedings of the American Association for Cancer Research (AACR).[9] These additional sets were added to increase the set of potentially relevant treatment information. Indeed, relevant literature articles can guide precision oncologists to the best-known treatment options for the patient's condition. Similarly to PM17 and PM18, the document set for TREC PM 2019 (PM19) consists of an updated set of $29,138,916$ MEDLINE abstracts. However, unlike PM17 and PM18, the ASCO and AACR abstracts were not included in the PM19 document set.

Clinical Trials

The Clinical Trials document set for PM17 and PM18 consists of a total of $241,006$ clinical trial descriptions, derived from ClinicalTrials.gov – a repository of clinical trials in the U.S. and abroad.[10]. When none of the available treatments is effective on oncology patients, the common recourse is to determine if any potential treatment is undergoing evaluation in a clinical trial. Precision oncology trials typically use a certain treatment for a certain disease with a specific genetic variant (or set of variants). Such trials can have complex inclusion and/or exclusion criteria that are challenging to match with automated systems. Again, the document set for PM19 consists of an updated set of $306,238$ clinical trial descriptions.

Topics

PM17, PM18, and PM19 contain, respectively, 30, 50, and 40 synthetic cases created by precision oncologists in 2017, 2018, and 2019. In 2017, synthetic cases contain four key elements in a semi-structured format: (1) disease (e.g. a type of cancer), (2) genetic variants (primarily present in tumors), (3) demographic information (e.g. age, gender), and (4) other factors which could impact certain treatment options. In 2018 and 2019, synthetic cases contain three of the four key elements used in 2017: (1) disease, (2) genetic variants, and (3) demographic information. In 2019, 30 of the 40 cases were created by precision oncologists, while the other 10 cases – unrelated to cancer – were based on the American College of Medical Genetics and Genomics (ACMG) recommendations.[11] These synthetic cases were

[8] https://www.asco.org/

[9] https://www.aacr.org/

[10] https://clinicaltrials.gov/

[11] https://www.ncbi.nlm.nih.gov/projects/dbvar/clingen/acmg.shtml

added to assess the relative difficulty of cancer search versus other disciplines requiring precision medicine.

2.1.4 Other Collections

Other than medical test collections, we also consider collections from different domains and in different languages. The domains considered are newswire and Web, whereas the languages are Italian, German, and Farsi. We rely on these collections, as well as medical ones, to investigate the impact that different domains and languages have on the performance of state-of-the-art IR systems.

Four of the newswire collections considered are subsets of the TIPSTER corpus [99]: Associated Press 88-89 (AP88-89), Financial Times (FT), LA Times (LA), and Wall Street Journal (WSJ) [101]. Then, we consider the Robust04 collection [232] – based on TIPSTER Disk 4/5 without the Congressional Record – and the New York Times (NY) collection [12], which consists of articles written and published by the *New York Times* between 1987 and 2007. We consider topics 50-200 from TREC 1-3 for AP88-89 and WSJ, topics 301-450 from TREC 6-8 for FT and LA, topics 301-450 and 601-700 from the TREC Robust Retrieval Track for Robust04, and the 50 topics from the TREC 2017 Common Core Track for NY.

For Italian, German, and Farsi languages we consider the newswire CLEF Italian (CLEF-IT), German (CLEF-DE), and Farsi (CLEF-FA) collections [60, 3]. CLEF collections present documents in different languages, but with common features: the same genre and period. CLEF-IT and CLEF-DE contain newspaper articles from 1994 to 1995, whereas CLEF-FA from 1996 to 2002. The Italian and German news agency dispatches are all gathered from the Swiss news agency and comprise the same corpus translated in different languages. CLEF-IT, CLEF-DE, and CLEF-FA contain, respectively, 90, 95, and 100 topics.

Finally, for the Web domain we consider the WT2g collection [104], a collection of documents crawled from the Web and used in the TREC 1999 Web Track. WT2g contains 50 topics.

2.2 Knowledge Resources

Knowledge resources represent a particular formal model (or view) of a mini-world of interest, be it a specific domain, subject area, or language. In other words, knowledge resources provide semantic information about the objects they store and the relationships that occur between them. Under the term "knowledge resource" lies a multitude of different and heterogeneous resources, which share common characteristics: factual knowledge, terminological consistency, and unambiguity [253]. However, the literature lacks a universally accepted definition of the different types of knowledge resources and their main constituents. For this reason, below, we provide a possible definition of the different types of knowledge resources and their main elements, which is functional to their understanding in the context of this thesis.

Among the different elements that can constitute a knowledge resource, the most important three to our work are: terms, concepts, and relations. **Terms** are words or phrases used to describe objects or to express concepts in a specific language or subject area. **Concepts** are elements of thought which represent abstract ideas or general notions. In a nutshell, concepts represent the meaning underlying the terms expressed through spoken or written language. Therefore, concepts are unambiguous and provide standardized definitions for terms expressing them. Knowledge resources typically contain preferred terms for concepts – which are used in the knowledge resource as the default terms to convey the meanings represented by the corresponding concepts. For instance, a knowledge resource could assign the preferred term "Malignant Neoplastic Disease" to the concept *Cancer*, rather than terms like "Cancer" or "Malignant Tumor". The choice of preferred terms is left to domain specialists. Finally, **relations** connect concepts and/or terms and can refer to any relationship of meaning between them. There are several types of semantic relations. For the purposes of this thesis, we focus on three categories: equivalence, hierarchical, and associative relations [22]. Equivalence relations connect terms through synonymy, quasi-synonymy, or lexical variant relationships. For example, the terms "Malignant Neoplastic Disease" and "Malignant Tumor" are synonyms for the concept *Cancer*. Hierarchical relations are divided into generic-specific (hyponymy-hypernymy) relationships and partitive (meronymy) relationships. Generic-specific relationships are also known as `is-a` relationships (e.g., "Asthma" `is-a` "Bronchial Disease"), whereas partitive relationships as `part-of` relationships (e.g., "Heart" `part-of` "Circulatory System"). Associative relations include sequential, spatial, temporal, and causal relationships – that is, those relationships that are not hierarchical nor equivalence relations. For a comprehensive and detailed description of these elements, we refer the reader to ISO standard 1087:2019.

Depending on the complexity of the underlying model and the relations considered, different types of knowledge resources can be defined. In our work, we focus on four types: nomenclatures, thesauri, ontologies, and knowledge bases. We present each type below, from the least to the most semantically expressive. A **nomenclature** (lit. "list of names") is a naming system for a given domain, formed according to strict linguistic rules. Nomenclatures are composed of terms collected by domain specialists and approved by scientific authorities. The purpose of nomenclatures is to standardize the use of the domain language to avoid ambiguity. In other words, the terms provided by nomenclatures act as preferred terms that univocally identify concepts within a given domain taxonomy – thus avoiding language inconsistencies and translation issues. Nomenclatures can be part of thesauri. A **thesaurus** is a controlled vocabulary and terminology, which denotes concepts and relations in a specific domain or subject area. It consists of systematized lists of synonyms, antonyms, and otherwise related terms. Terms are grouped in a taxonomy of concepts through the use of hierarchical relations. Thesauri use preferred terms to refer unambiguously to concepts, avoiding the need to impose additional model constraints. Thesauri can form part of ontologies. An **ontology** is a semantic data model defining the types of concepts and objects that exist in a given domain or subject area, as well as the properties that can be used to describe them. In 1993, Gruber originally defined the notion of ontology as an "explicit specification of a conceptualization" [91]. In plain words, an ontology is a mean to formally model the structure of a system – that is, the relevant concepts and relations that emerge from its observation and are useful to a specific purpose [92]. The backbone of an ontology consists in hierarchical relations between concepts, and it can be extended with different relations (e.g., equivalence and associative relations) reflecting the specific mini-world of interest. Ontologies are used by knowledge bases as the underlying semantic data model to which data instances must comply with. A **knowledge base** is a database system that uses semantic data models to store and retrieve knowledge. Knowledge bases aim to link and integrate all the available knowledge sources for a specific domain or subject area, including explicit knowledge – stored within existing information systems – and implicit knowledge – derived from the practical experience and understanding of domain specialists. Put simply, a knowledge base can be seen as a collection of explicit and implicit knowledge related to the concepts – and the relations between concepts – of a specific domain or subject area.

Below, we present the actual knowledge resources used in our work.

2.2.1 Systematized Nomenclature of Medicine - Clinical Terms

The Systematized Nomenclature of Medicine - Clinical Terms (SNOMED CT) [66] is a logic-based health care thesaurus, which originated from the Systematized Nomenclature of Pathology (SNOP).[12] It is the most comprehensive, multilingual clinical healthcare terminology worldwide. The main objective of SNOMED CT is to enable users to encode different kinds of health information in a standardized way, thus improving patient care. SNOMED CT presents a multi-hierarchical and multi-axial structure (i.e., concepts can have more than one superordinate concept) and includes three components: concepts, terms, and relations.

[12] http://www.snomed.org/

Concepts are organized from the most general to the most specific through hierarchical, `is-a` relationships. Then, associative relationships connect concepts whose meaning is related in non-hierarchical ways. These relationships provide formal definitions and properties, like: `causative agent`, `finding site`, `pathological process`, etc. Each concept has a unique concept code (or ID) that identifies the clinical terms used to designate that concept. The terms describing concepts can be divided into fully specified names, preferred terms, and synonyms.

2.2.2 Medical Subject Headings

The Medical Subject Headings (MeSH) thesaurus [146] is a controlled and hierarchically-organized thesaurus used for indexing, cataloging, and searching biomedical and health-related information.[13] MeSH includes the subject headings appearing in MEDLINE, the National Library of Medicine (NLM) Catalog, and other NLM databases. MeSH is available in several languages and presents a tree structure, from the most general concept to the most specific. The terms describing concepts can be divided into preferred terms and synonyms. MeSH is constantly updated by domain specialists in various areas. Each year, hundreds of new concepts are added, and thousands of modifications are made.

2.2.3 National Cancer Institute Thesaurus

The National Cancer Institute (NCI) thesaurus [205] is the NCI's reference thesaurus, covering areas of basic and clinical science and built with the goal of facilitating translational research in cancer.[14] It contains terms, concepts and relations. The concepts are partitioned in subdomains, which includes, among others, diseases, drugs, genes, anatomy, and biological processes – all with a cancer-centric focus in content. Each concept presents a preferred name and a list of synonyms, as well as annotations such as textual definitions and (optional) references to external sources. Besides, concepts are defined by their relationships to other concepts.

2.2.4 Unified Medical Language System Metathesaurus

The Unified Medical Language System (UMLS) metathesaurus [29] is a large, multi-purpose, and multi-lingual knowledge base that contains information about biomedical and health related concepts, their name variants, and the relationships among them.[15] The metathesaurus

[13] https://www.ncbi.nlm.nih.gov/mesh/

[14] https://ncithesaurus.nci.nih.gov/ncitbrowser/

[15] https://www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/

is built from different thesauri, classifications, ontologies, code sets, and lists of controlled terms used in patient care, health services, biomedical literature, etc. All the concepts in the metathesaurus are assigned a unique and permanent Concept Unique Identifier (CUI), along with a preferred name and at least one semantic type from the UMLS Semantic Network.[16] Semantic types provide a consistent categorization of all the concepts in the metathesaurus at the relatively general level represented in the Semantic Network. The metathesaurus includes equivalence, hierarchical, and associative relationships between concepts. Most of these relationships come from individual source vocabularies, but some are added by NLM during metathesaurus construction. The UMLS metathesaurus is updated twice per year.

2.2.5 Cancer Biomarkers Database

The Cancer Biomarkers database [219] is a knowledge base containing information on genomic biomarkers of response (sensitivity, resistance, or toxicity) to different drugs across different types of cancer, which extends a previous collection of genomic biomarkers of anti-cancer drug response [64].[17] Negative results of clinical trials are also included in the database. Biomarkers are organized according to the level of clinical evidence supporting each one, ranging from results of pre-clinical data, case reports, and clinical trials in early and late phases to standard-of-care guidelines. The Cancer Biomarkers database is updated by medical oncologists and cancer genomics experts.

[16] https://semanticnetwork.nlm.nih.gov/

[17] https://www.cancergenomeinterpreter.org/biomarkers/

Chapter 3

Background

The semantic gap is a long-standing problem in IR. For this reason, a wide variety of different approaches have been proposed, designed, and developed to address it. Over the years, two main lines of work have emerged to bridge the semantic gap between queries and documents: (i) the use of external knowledge resources to enhance bag-of-words query and document representations, and (ii) the use of semantic models to perform matching between the latent representations of queries and documents.

The work we present in this thesis mainly belongs to the two lines of research described above. Hence, in this chapter, we provide the necessary background on the different approaches that have been proposed in these two lines of work. Other than that, we also present those models and techniques that have been used by IR systems to address the semantic gap, but that have been proposed in other fields and/or for other tasks – i.e., that have not been explicitly designed for IR. In this way, the reader can have a comprehensive view of all the components that lead to the realization of an IR system designed to address the semantic gap.

The rest of this chapter is organized as follows. In Section 3.1, we review the Information Extraction (IE) tools we employ in our work to enhance retrieval models with external knowledge. In Section 3.2, we review the different approaches that have been developed to integrate external knowledge in lexical models. In Section 3.3, we discuss the advances in neural representation learning – on which many of the models from Sections 3.4 and 3.5 are based. In Section 3.4 we review both traditional and neural semantic models. Finally, in Section 3.5, we present the (recent) works on knowledge-enhanced semantic models. When necessary, we highlight similarities and differences between our work and the literature, also providing pointers to the following chapters of this thesis.

3.1 Information Extraction Tools

IE regards the automatic extraction of knowledge from unstructured, semi-structured, or structured machine-readable text. In other words, IE makes it possible to extract information from a wide variety of textual sources, such as text documents, Web pages, tables, etc. Among the different IE tasks, Named Entity Recognition (NER) and Entity Linking (EL) are of particular interest for IR. NER seeks to identify and classify named entities – i.e., real-word objects that can be denoted with a proper name – mentioned in unstructured or semi-structured text into a set of pre-defined categories, such as persons, organizations, locations, medical codes, etc. Compared to NER, EL takes a further step by linking named entities to corresponding unique entities contained in an external knowledge resource. In a nutshell, while NER identifies the occurrence of a named entity in text, EL identifies which unique entity that occurrence is.

Thus, IE techniques – and in particular EL tools – can be used on queries and documents to identify entities that can increase the representative power of retrieval models, helping them to bridge the semantic gap. Also, the use of EL tools provides access, through the linked entities, to the underlying knowledge resources and their relational semantics that, again, can be used to enhance retrieval models.

While a wide variety of off-the-shelf EL tools have been used in IR [15, 75, 198, 160, 207], in this section, we review only those we employed in our work, namely MetaMap [15] (used in Chapter 5) and QuickUMLS [207] (used in Chapter 6).

MetaMap [15] is a state-of-the-art tool for recognizing UMLS [29] concepts within biomedical text. It was developed more than twenty years ago, and has continued to evolve and improve since then [16]. MetaMap is guided by linguistic, rather than statistics, principles which provide a flexible architecture to perform concept mapping strategies. Given an input text, MetaMap performs the following lexical/syntactic analysis. First, it performs tokenization, sentence boundary detection, and acronym/abbreviation identification. Then, it applies part-of-speech tagging, a lexical lookup of input words into a domain-specific lexicon, and a syntactic analysis in which phrases and their lexical heads are identified by a domain-specific parser. Each phrase found by this analysis undergoes the following process. First, a variant generation step is performed, where variants of all phrase words are identified (by table lookup). Then, a candidate identification step runs, where UMLS concept labels are compared against the input phrase to evaluate how well they match it. Finally, the mapping construction step evaluates the candidate concepts found in the previous step and produces a final result that best matches the input phrase. Optionally, MetaMap can also perform Word Sense Disambiguation (WSD) to prioritize the candidate concept that is most semantically consistent with the surrounding text of the target phrase.

Although MetMap is one of the most effective and widely-used concept extraction tools for medical literature, it is hampered by one main limitation: it does not scale efficiently to large collections. To overcome this limitation, Soldaini and Goharian [207] developed QuickUMLS, a fast, unsupervised, approximate dictionary matching tool for medical concept extraction based on UMLS [29]. Compared to MetaMap, QuickUMLS achieves similar precision and recall values but requires significantly less time to run. Given an input text, QuickUMLS efficiently generates, for each word, all the possible sequences that stem from it up to a fixed maximum length. Then, QuickUMLS adopts a series of heuristics to determine whether a given sequence can be regarded as valid or not. If the sequence is valid, then QuickUMLS performs approximate dictionary matching to find concept labels within UMLS that are similar to the target sequence. Once the set of all possible matching labels is determined for the entire input text, QuickUMLS selects the most appropriate subset such that there is no overlap between the extracted concepts.

3.2 Knowledge-Enhanced Lexical Models

Enhancing lexical models through external knowledge to improve retrieval effectiveness dates back to the early years of IR. In 1965, Rocchio and Salton [194] proposed several search optimization procedures to help users in formulating effective queries. The proposed methods can be divided into two groups: Vocabulary Feedback (VF) and Relevance Feedback (RF). The methods that belong to VF rely on external knowledge resources to display terms to the user that are related in various ways to those of the original query. Given a user search, the system displays the statistics related to the query terms in the document collection, along with the list of underlying concepts associated with those terms in the reference thesaurus. In this way, the user can decide to reformulate its original query based on the related terms contained in the external resource – selecting broader terms if the query is over-specific and narrower terms if it is under-specific instead. Hence, the external knowledge resource helps users to manually reformulate their queries.

Although effective, the methods belonging to VF require a considerable effort from the user, who needs to control both what the system displays and returns. As an attempt to shift (part of) the burden from the user to the system, RF methods have been proposed [193]. The idea behind RF methods is to involve the user in the retrieval process with the objective of improving the (final) ranking list. Basically, the user examines some of the retrieved documents and classifies them as relevant or non-relevant depending on its information need. These relevance judgments are then returned to the system, which uses them to adjust the original query in such a way that query terms contained within relevant documents are promoted (e.g., by increasing their weight) and query terms contained within non-relevant are similarly demoted. The relevance feedback process can occur multiple times before the user satisfies its information need. In other words, it is an iterative query refinement process that leads users to improve the understanding of their (evolving) information need [23].

Feedback methods have largely impacted IR and have been widely used since then. Attar and Fraenkel [18], and Croft and Harper [52] first proposed Pseudo Relevance Feedback (PRF) methods as a way to fully automatize RF to obtain expansion terms [18] or perform query reweighting [52]. Put simply, PRF assumes the top-ranked documents, previously retrieved by an IR system, to be relevant and performs query reweighting and/or expansion based on this information. At the same time, methods relying on external knowledge resources to enhance bag-of-words representations have evolved from the VF paradigm into fully automatic approaches. We can divide these methods in three categories: (i) methods that integrate external knowledge in the indexing stage [2], (ii) methods that integrate external knowledge in the retrieval stage [230], and (iii) methods that integrate external knowledge in both indexing and retrieval stages [229, 161].

In this section, we focus on knowledge-enhanced methods as they explicitly address the semantic gap between queries and documents. Indeed, the relational information contained within authoritative and structured knowledge resources can be used to model linguistic features related to the semantic gap (e.g., synonymy and polysemy) and enhance bag-of-words representations used by lexical models. Besides, knowledge-enhanced lexical models are often based on pre-retrieval techniques – that is, on techniques that operate during indexing or before the query is issued. Thus, enhancing lexical models through external knowledge also impacts post-retrieval techniques like PRF. In Chapter 6, we confirm this cascading effect using semantic models as well – showing that knowledge-enhanced semantic models grasp different signals than lexical models and retrieve documents in top positions that are effective in providing expansion terms for PRF based lexical models (see Section 6.8).

Below, we review knowledge-enhanced methods for each of the three considered categories.

3.2.1 Knowledge-Enhanced Document Representations

The use of external knowledge resources to enhance bag-of-words document representations at the indexing stage can be regarded as a type of document expansion [203]. Document expansion addresses the semantic gap between documents and queries by enriching documents with related terms. In this context, only few approaches have been proposed that use external resources to expand documents [2, 1]. Agirre et al. [2] developed a document expansion method that relies on WordNet [164] – a lexical database of semantic relations between words – to identify related concepts and words.[1] Given a document, a random walk algorithm is performed over the WordNet graph to rank concepts that are closely related to document words. Compared to WSD approaches, that replace words with their senses, the random walk approach can discover relevant concepts even if they are not explicitly mentioned within the document. Once identified, expansion words are indexed separately from those originally contained within documents, and the two indexes are linearly combined in the retrieval stage. The experimental results showed that combining the indexes proves effective and provides state-of-the-art performances in different test collections. In particular, the authors found that document expansion is highly effective for short documents. The same document expansion strategy was then employed by Agirre et al. [1] in the context of cross-lingual passage retrieval. However, the authors found that the use of document expansion was not effective in this cross-lingual task, probably due to limitations in the translation process.

3.2.2 Knowledge-Enhanced Query Representations

Compared to document expansion, a far more popular approach is to enhance query representations. Among the models integrating external knowledge in the retrieval stage, Voorhees [230] proposed one of the first approaches based on WordNet [164]. The approach performs query expansion based on the relational information contained within WordNet. First, query terms are manually annotated with concepts from WordNet – also known as synonym sets (or synsets) – to avoid introducing noise in the disambiguation process. Then, expansion terms are automatically identified by following semantic relationships within WordNet like synonymy or hierarchical relations. Finally, expanded queries are used to perform retrieval against the document collection. The experimental results showed that the proposed query expansion has a negligible impact when original queries are long representative descriptions of the underlying information need. On the other hand, short and less representative queries significantly benefit from the expansion process proposed. However, due to the use of manual annotations for query concepts, the results represent an upper bound for the performances of a fully automated system.

Along the same lines, Navigli and Velardi [167] proposed a query expansion approach, based on WordNet, for Web retrieval. Compared to Voorhees [230], however, Navigli and Velardi developed a semiautomatic WSD technique to identify WordNet concepts within queries. Then, relying on the relational and semantic information contained within WordNet, the authors performed different query expansions based on synonymy and hierarchical

[1] https://wordnet.princeton.edu/

relations, as well as textual information contained within synset descriptions. The results showed that query expansions based on synonyms or hyperonyms have a limited impact on retrieval performances for Web collections. On the other hand, words co-occurring in the same sysnet descriptions or belonging to the same semantic domain are effective expansion terms. Like Voorhees [230], Navigli and Velardi also concluded that query expansion is best suited to short queries.

The methods proposed by Voorhees [230] and Navigli and Velardi [167] are pre-retrieval approaches. On the other hand, Pal et al. [174] proposed a post-retrieval method based on WordNet. Expansion terms are extracted from pseudo-relevant documents (i.e., feedback documents) as in standard PRF based expansion [18, 33, 241, 121], but the weight of these terms also depends on the similarity between their WordNet definitions and those of the query terms. Experimental results on TREC collections showed that the proposed method outperforms previous effective WordNet-based expansion methods [73].

The use of external knowledge to enhance query representations have been investigated also for domain-specific applications. In particular, knowledge-enhanced query reformulation techniques have been successfully employed in medical retrieval. Srinivasan [213] investigated different knowledge-enhanced query expansions within a PRF framework. The three alternative expansions are based on: MeSH [146] terms, corpus terms, and both MeSH and corpus terms. Experiments on a MEDLINE test collection showed that query expansions based on MeSH terms significantly outperform the expansion based on corpus terms only. Afterwards, Srinivasan [212] performed an evaluation on knowledge-enhanced pre- and post-retrieval query expansion strategies on the same MEDLINE test collection used in [213]. The expansion strategies considered are: a (pre-retrieval) MeSH expansion, a (post-retrieval) PRF expansion, and the combination of both. The results showed that expanding the query with terms from MeSH proves effective, but the combination of both pre- and post-retrieval techniques is better. Aronson and Rindflesch [17] proposed to expand queries using MetaMap [15]. Given a query, MetaMap maps the query text to the concepts within the UMLS metathesaurus [29]. Then, the query is expanded with both the surface forms and the preferred names of the identified UMLS concepts. The authors compared the proposed MetaMap-based query expansion with the pre-retrieval methods developed by Srinivasan [213, 212]. The results highlighted that the MetaMap-based query expansion achieves the largest improvements – close to those obtained using the combination of pre- and post-retrieval techniques developed by Srinivasan [212]. As opposed to Aronson and Rindflesch [17], who performed automatic query expansion relying on MetaMap [15], Hersh et al. [110] expanded queries with terms manually selected from the UMLS metathesaurus [29]. Selected terms present synonymy, hierarchical, or other relationships with query terms. The experimental results showed that, although being effective on specific queries, all the expansion techniques based on the selected terms degrade average performances.

In the context of genomics, where the objective is to retrieve biomedical articles describing how genes contribute to diseases in living organisms, Stokes et al. [214] explored the criteria required to perform effective query expansions. Based on the results of the TREC Genomics 2006 Track [108],[2] the authors developed a query expansion framework they used to evaluate the effectiveness of the different expansion approaches employed by TREC Genomics 2006 Track participants. The experimental results showed that the choice of the (lexical) ranking model is the factor that affects retrieval performances the most. Once an effective ranking model is found, query expansion techniques based on domain-specific knowledge resources provide the largest performance improvements.

In the context of Clinical Decision Support (CDS), where the objective is to retrieve biomedical articles relevant for answering clinical questions about medical records, Soldaini et al. [206] investigated the effectiveness of query expansion and reduction techniques based on medical knowledge resources, as well as general-purpose IR techniques. Regarding query reduction techniques, the authors removed those query terms that are not related to any UMLS concept or Wikipedia health-related entry.[3] Moreover, they also relied on query quality predictors [135] to identify effective sub-queries that can replace the original query. Regarding query expansion techniques, the authors relied on both pre- and post-retrieval approaches. For pre-retrieval, they identified UMLS concepts within queries and performed the expansion by adding the UMLS preferred terms associated to those concepts. Besides, to prevent topic drift – a phenomenon which often occurs when the query is expanded with terms that are not pertinent to the information need [228] – Soldaini et al. considered only those UMLS concepts related to drugs, diseases, and findings. On the other hand, for post-retrieval, they introduced a new method that combines a domain-specific approach with PRF. The experimental results showed that query reduction based on Wikipedia health-related entries proves effective, but the post-retrieval method combining a domain-specific approach with PRF performs best. The approach was then extended by Soldaini et al. [208], who improved the method to select expansion terms – leading to significant improvements over the considered baselines in TREC CDS 2014 [188] and 2015 [189] collections.

In the context of Precision Medicine (PM), where the objective is to provide useful precision medicine-related information to clinicians treating cancer patients, query expansion and reduction techniques have been proven highly effective. In the TREC PM 2017 Track [186], López-García et al. [148] relied on various domain-specific knowledge resources to perform

[2] https://dmice.ohsu.edu/trec-gen/

[3] https://en.wikipedia.org/

disease and gene expansions. Given a query, the disease and gene fields were expanded using all the synonyms for the identified concepts. Then, different tuning strategies were applied to the expanded queries with the objective of diversifying the importance of the terms coming from different sources. Among the various runs submitted to TREC PM 2017, those generated from the approach proposed by López-García et al. belong to the top 3 performing runs in the Scientific Literature task. Building on the work by López-García et al. [148], Oleynik et al. [172] participated in the TREC PM 2018 Track [185] and developed various hand-crafted rules to mitigate the effect of detrimental information contained within either documents or queries. First, they defined, for both Scientific Literature and Clinical Trials tasks, an exact-match query clause that excludes the possibility of matching "non-melanoma" when the query contains the word "melanoma". Then, they proposed, for the Clinical Trials task only, two additional rules. The first rule regards the expansion of those queries that do not mention any kind of blood cancer (e.g., "lymphoma" or "leukemia") with the term "solid". Indeed, as pointed out also by Goodwin et al. [90], a large part of (relevant) clinical trials does not mention the exact topic disease, but rather adopts an umbrella term like "solid tumor". On the other hand, the second rule regards the reduction of the gene information from the exact gene to the gene family (e.g., from "PIK3CA" to "PIK"). The reduction process aims to mitigate the over-specificity of topics, as the information contained within topics can be too specific compared to that contained within target documents. As a side note, this second rule does not rely on any external and authoritative knowledge resource but rather it applies a simple and straightforward regular expression. Among the different runs submitted to TREC PM 2018, those generated from the approach proposed by Oleynik et al. [172] belong to the top 10 performing runs in both tasks for all the considered measures. Following in the footsteps of Oleynik et al. [172], Faessler et al. [71] employed the approach developed by López-García et al. [148] – and then revised by Oleynik et al. [172] – in the TREC PM 2019 Track [187], achieving top performances for both Scientific Literature and Clinical Trials tasks.

The work we present in Chapter 5 follows an iterative process similar to that reviewed in the context of precision medicine. First, we have conducted a preliminary study on the TREC PM 2018 Clinical Trials task (see Section 5.1), where we proposed a procedure to expand queries iteratively – relying on medical knowledge resources – and filter out trials for which the patient is not eligible. We started with TREC PM 2018 as it provides the most representative collection for precision medicine – i.e., the one with the largest number of topics related to cancer cases. Then, driven by the results of this preliminary study, we have deepened the analysis of knowledge-enhanced query reformulations, and we have extended it to both scientific literature and clinical trials retrieval (see Section 5.2). The evaluation, performed on TREC PM 2017 and 2018, showed the effectiveness of the proposed query reformulations in both test collections. Building on these findings, we have employed our approach for the TREC PM 2019 Track [187] (see Section 5.3) – where the experimental results validated the effectiveness of our (tested) query reformulations for retrieving relevant clinical trials. A final analysis stemmed from these works, where we have focused on the effectiveness of the proposed query reformulations for clinical trials retrieval (see Section 5.4). The outcomes of this analysis highlighted a subset of query reformulations effective across all the three editions.

Beyond medical applications, Fu et al. [83] proposed query expansion techniques based on both domain and geographical ontologies to better answer queries that involve spatial terms and relationships. The authors relied on different factors to expand queries, including spacial terms and relationships – contained within geographical ontologies – and semantic concepts – contained within domain ontologies. Expanded queries showed to improve retrieval performances.

3.2.3 Knowledge-Enhanced Query and Document Representations

One of the first approaches integrating external knowledge in both indexing and retrieval stages has been proposed by Voorhees [229], who developed an automatic indexing procedure based on WordNet [164]. The proposed indexing relies on the `is-a` relationships contained within WordNet and the set of nouns within documents and queries to select a word sense for each polysemous noun. This means that the resulting index stores word senses rather than words for polysemous words. Voorhees evaluated the effectiveness of this semantic index by comparing retrieval performances obtained with it or with a traditional index, which stores word stems instead of word senses [149]. The experimental results showed that relying on word stems provides better performances than word senses – although the semantic index proves effective on some queries. The main cause of this degradation lied in the difficulty of disambiguating senses in short queries. From this, Voorhees drew two conclusions: 1) `is-a` relationships are not sufficient to correctly disambiguate nouns, and 2) missing correct matches because of incorrect disambiguation has a worse impact on retrieval performance than making spurious matches.

Along the same lines, Mihalcea and Moldovan [161] developed a semantic indexing technique that first applies WSD based on WordNet [164] senses (i.e., synsets) on documents, and then performs indexing separately over words and disambiguated senses. Once the index is ready, an input query is first disambiguated and then matched against the semantic index. Compared to Voorhees [229], who relied on `is-a` relationships to disambiguate word senses, Mihalcea and Moldovan opted for a semi-complete disambiguation algorithm – capable of disambiguating about 55% of the nouns and verbs with an accuracy greater than 92%. The results showed that indexing by both words and senses improves retrieval performance. In particular, Mihalcea and Moldovan were the first to apply a WSD algorithm for free-text to IR document collections and improve retrieval performances.

Also in this case, many approaches have been developed for the medical domain. Dinh and Tamine [65] developed an approach that combines document expansion and query expansion to improve retrieval effectiveness in the genomics domain. The authors adopted different external knowledge resources – such as MeSH [146] and SNOMED CT [66], among the others – to extract concepts from documents. Then, relying on rank fusion techniques [200], the concepts extracted with different resources are used to select the best candidate terms for document expansion. Finally, queries are expanded using PRF over the knowledge-enhanced document collection. The experiments performed on the TREC Genomics 2004 [106] and 2005 [109] test collections showed the effectiveness of the proposed approach. Rather than expanding documents and queries, Limsopatham et al. [142] focused on shifting bag-of-words representations from words to concepts. Relying on MetaMap [15] to identify UMLS [29] concepts within documents and queries, they developed concept-based representations related to the four aspects of the medical decision criteria: symptoms, diagnostic tests, diagnoses, and treatments. These aspects represent the necessary information health practicioners need to assist their patients. The authors extended the proposed approach in [143], where they expanded queries in two ways: by inferring additional semantic relations from external knowledge resources, and by extracting informative concepts from top-ranked medical records. Experimental results showed the effectiveness of the approach in modeling implicit knowledge for medical records retrieval. Koopman et al. [132] have also exploited concept-based representations to perform medical records retrieval. The proposed approach builds document and query representations relying on SNOMED CT [66] concepts and exploits SNOMED CT `is-a` relationships to weight the documents that contain concepts subsumed by those in the query. On the other hand, Agosti et al. [7] investigated the use of semantic relations to improve bag-of-words representations. In other words, they evaluated how semantic relations – identified between concepts extracted from medical documents and linked to a reference knowledge resource – can be leveraged to retrieve medical literature for CDS tasks. The authors proposed two methods to identify relations within queries and documents: a rule-based method and a learning-based method. Then, they compared the effectiveness of bag-of-words, bag-of-concepts, and bag-of-relations representations when used to perform retrieval. From the experiments performed on the OHSUMED collection [107], they found that relations – when pertinent to the information need – outperform the contribution provided by words or concepts. However, the amount of queries where semantic relations provide effective results is limited.

3.3 Representation Learning

Representation learning aims to learn representations of input data – by transforming such data or extracting features from it – to perform tasks like classification, prediction, or retrieval [24]. Representations can be learned in different ways. For instance, probabilistic models learn representations that capture the posterior distribution of the underlying explanatory features given the observed input. On the other hand, deep learning models learn representations by composing multiple non-linear transformations of the input data to obtain abstract and effective representations. In other words, learned representations depend on the model adopted and on the input data considered.

In this section, we review neural representation models developed for text data. Within this context, many applications of representation learning have been proposed. Historically, distributed word representations were introduced by Hinton [111] to model structural relationships between concepts, and then first developed by Bengio et al. [26] for statistical language modeling. The models we review stem from these pioneering works and are all based on learning distributed representations for textual units – be them words, phrases, sentences, or even documents – called embeddings. Knowing the different approaches developed, together with their strengths and weaknesses for the tasks they have been originally proposed for, helps to understand the impact that representation learning can have – and already has – in Information Retrieval (IR).

We divide neural representation learning into two main categories: corpus-driven, where the representations are learned relying solely on the text corpus, and knowledge-enhanced, where the representations are learned relying on the text corpus and external knowledge resource(s).

3.3.1 Corpus-Driven Representation Learning

Since the conception of neural language models [26], building low-dimensional representations of words from large corpora has gained increasing attention in the NLP community. The word2vec models proposed by Mikolov et al. [162] are based on the Distributional Hypothesis [102]. They use the local co-occurrences of words to learn embedded representations of words. In particular, the Continuous Bag-Of-Words (CBOW) architecture predicts a target word by maximizing the log-likelihood of its context words within a fixed-size window, whereas the skip-gram architecture predicts the context words within a fixed-size window given the target word. Conversely, the Global Vector (GloVe) model [179] learns embedded representations of words based on their global co-occurrence. However, by assigning a distinct representation to each word, word2vec and GloVe ignore the morphology of words. To overcome this limitation, Bojanowski et al. proposed fasttext [30], a new approach based on skip-gram where each word is represented as a bag of character n-grams. fasttext associates a vector representation to each character n-gram, so that words are represented as the sum of these representations. In this way, fasttext can handle out-of-vocabulary words by simply averaging the vector representations of the n-grams composing such words. The results showed that morphological information significantly improves the effectiveness on syntactic tasks, but it does not help for semantic tasks – where it even degrades performances.

More recently, contextual neural language models have been proposed to overcome the lack of contextualization of traditional word embeddings. Contextual neural language models generate different word representations for the same word given the context in which the word occurs. Context2vec [159] learns a generic context embedding function using a bidirectional Long Short-Term Memory (LSTM) architecture. Embeddings from Language Models (ELMo) [180] introduces deep contextualized word representations that model both complex characteristics of word use (e.g., syntax and semantics) and how these uses vary across linguistic contexts (i.e., polysemy). The word vectors derive from the internal states of a deep bidirectional language model pretrained on a large corpus. Similarly, Bidirectional Encoder Representations for Transformers (BERT) [59] models complex characteristics relying on self-attention layers from Transformer networks [227]. Despite being very powerful, contextual neural language models have complex architectures and high computational costs [125, 32], which make even their fine-tuning complicated on large-scale collections like those typically used in IR. For this reason, contextual neural language models have been mainly used in IR as supervised approaches to perform reranking [246, 171, 50].

Other than learning word representations, methods that learn distributed representations of sentences, paragraphs, or documents have also been proposed. Kenter et al. [126] proposed the Siamese CBOW model, which takes inspiration from the CBOW model to learn a target sentence from its surrounding (context) sentences. Similarly, the Skip-thought model [129] learns sentence representations by predicting context sentences from the target sentence. As an extension to word2vec, Le and Mikolov [138] proposed the doc2vec models. Both doc2vec Distributed Bag-Of-Words (DBOW) and Distributed Memory (DM) architectures jointly learn document and word representations within the same vector space. The DBOW architecture mimics the behavior of word2vec skip-gram architecture, whereas the DM architecture mimics the behavior of word2vec CBOW architecture. Chen [38] presented the Document Vector through Corruption (Doc2VecC), an efficient document representation learning framework. Doc2VecC represents each document as a simple average of word embeddings and it ensures that word representations capture the semantic meanings of the documents during learning. The corruption component introduces a data-dependent regularization that favors informative or rare words and forces the embeddings of common and non-discriminative words to be close to zero. The advances brought by Transformer-based neural language models have led Cohan et al. [45] to propose Scientific Paper Embeddings using Citation-informed TransformERs (SPECTER). SPECTER generates document-level embeddings by pretraining a contextual neural language model on document-level relatedness signals obtained from the citation graph. The experimental results showed that SPECTER embeddings outperform competitive baselines on a variety of document-level tasks, ranging from citation prediction to document classification and recommendation.

3.3.2 Knowledge-Enhanced Representation Learning

Distributed representations of words capture the latent relations existing between words by relying only on the corpus as a knowledge resource. In the past few years, several approaches that combine corpus-based information with external knowledge resources to enhance word, sentence, or document representations have emerged. These approaches have been mainly developed to address polysemy and synonymy.

Faruqui et al. [74] proposed the retrofitted word2vec (rword2vec). rword2vec retrofits word embeddings using the relational information contained within semantic lexicons. The method forces words connected in the lexicon to have similar representations by minimizing both the distance of each word with its connected words in the lexicon and the distance with its pre-trained representation – namely, the distributed representation obtained with word2vec. Similarly, the counter-fitting method [166] refines distributed word representations relying on both synonymy and antonymy constraints. Johansson and Pina [123] proposed a retrofitting approach to address polysemy. First, the approach decomposes the vectors of polysemous words into a convex combination of sense vectors; secondly, it keeps sense vectors similar to those of the neighboring senses in the knowledge resource. Rather than integrating relational constraints directly into the learning objective, Glavas and Vulić [87] transformed external lexical semantic relations into training examples which are used to learn an explicit retrofitting model. The model learns a global specialization function that specializes the vectors of words unobserved during training too.

Yu and Dredze [250] proposed a representation model that combines the objective function of neural language models with prior knowledge from external resources to learn improved lexical-semantic word representations. The RC-NET [240] framework exploits both relational and categorical knowledge to produce knowledge-enhanced word representations. In particular, relational and categorical knowledge are encoded through different regularization functions and combined with the original objective of the word2vec skip-gram architecture. Yamada et al. [242] proposed to learn separate vector spaces for word and concepts and then align them through an anchor-context model which exploits anchors, contained within a knowledge resource, and their context words. The learned word and concept representations were used to perform EL. Iacobacci et al. [119] proposed an approach to improve semantic similarity that shifts from the word-level to the sense-level by leveraging knowledge from an external resource. Similarly, Mancini et al. [156] proposed a model that jointly learns word and sense representations. The model exploits corpus-based information and knowledge from external resources to produce a unified vector space of word and sense embeddings. Conversely, Cheng et al. [40] proposed a framework to generate context-aware text representations without diving into the sense space. The proposed framework projects both words and concepts into the same vector space and produces contextual word representations preserving the uniqueness among words while reflecting their context-appropriate meanings. Devine et al. [56] proposed to measure semantic similarity between medical concepts using a variation of the neural language models that learns on concepts from a knowledge resource and extracted from a corpus.

Regarding contextual neural language models, ERNIE models [217, 255] – namely, Enhanced Representation through kNowledge IntEgration [217] and Enhanced language RepresentatioN with Informative Entities [255] – extend BERT by incorporating knowledge resources in the learning process. To the best of our knowledge, ERNIE models have not been used in IR yet.

Beyond word-level representations, Sinoara et al. [204] proposed an approach that relies on WSD tools and embedded representations of words and word senses to represent documents. The constructed document representations were then used for text classification. Choi et al. [41] proposed a model to learn representations for medical concepts and visits. Given the sequential nature that medical visits possess for each patient, the model treats the document context – i.e., the medical visit – as a temporal feature.

3.4 Semantic Models

Semantic models were introduced to overcome the limitations of lexical matching related to the semantic gap [140]. In fact, given query and document bag-of-words representations, lexical models fail to retrieve relevant documents that express the query concepts with different words – as they compute the relevance score using heuristics defined over the lexical overlap between query and document representations. On the other hand, semantic models rely on low-dimensional representations to perform semantic matching between queries and documents in a latent semantic space. Unlike lexical matching, semantic matching computes the similarity score between two elements – which can be a pair of words, or a query/document pair – as the distance, under a given metric, between their representations projected into a common low-dimensional space. In this way, semantic models identify similarities at the semantic level where lexical models often fail.

Semantic models can exploit different signals other than semantic ones, and the use of latent representations within them vary from model to model. In this section, we review traditional semantic models and neural IR models. Traditional semantic models represent those pioneering works that first relied on low-dimensional latent representations to address the semantic gap between queries and documents in IR. On the other hand, neural IR models fall within the current deep learning wave and represent those approaches that rely on (deep) neural networks to perform retrieval.[4]

3.4.1 Traditional Semantic Models

Semantic models have been used for decades in IR as a means to mitigate the semantic gap between queries and documents and retrieve relevant documents that lexical models fail to discover. Among them, the first model proposed was Latent Semantic Indexing (LSI) [57]. LSI aims to find a data mapping that provides information beyond the lexical level and reveals semantic relations between documents and queries. In LSI, high-dimensional count vectors, such as those arising in document vector space representations [196], are mapped to a lower dimensional representation within a latent semantic space. In other words, LSI replaces the original vector space representation of documents and queries with a low-dimensional representation in the latent space. Specifically, LSI leverages Singular Value Decomposition (SVD) to decompose a large term-by-document matrix into a set of orthogonal factors from which the original matrix can be approximated by linear combination. Thus, documents are represented as vectors of factor weights and queries as pseudo-document vectors formed from weighted combination of terms. When performing retrieval, document and query vectors are matched using cosine similarity and documents are returned in decreasing order of their cosine value.

[4] The beginning of the deep learning era is commonly traced back to the work of Hinton et al. [112] on deep belief networks, where the authors first introduced the possibility of training a deep neural network by layer-wise training.

As an alternative to LSI, Hofmann proposed the probabilistic LSI (pLSI) [113]. Compared to LSI, which stems from linear algebra and performs SVD on co-occurrence matrices, pLSI models the probability of each co-occurrence as a mixture of conditionally independent multinomial distributions. In other words, pLSI models each word in a document as a sample from a mixture model, where the mixture components are multinomial random variables that can be viewed as representations of topics. Thus, each word is generated from a single topic, and different words in a document can be generated from different topics. Each document is represented as a list of mixing proportions for these mixture components and then reduced to a probability distribution on a fixed set of topics. This distribution represents the document latent description. Regarding query representations, pLSI keeps query factors fixed and adapts only mixing proportions.

A significant step forward over pLSI in probabilistic text modeling was Latent Dirichlet Allocation (LDA) [28], a generative probabilistic model for text corpora. LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities. The topic probabilities provide an explicit representation of a document or a query. Given the similarity of LDA with language models [181, 27, 254], its representations can be used in the IR language modeling framework [238].

Even though traditional semantic models offer an interesting framework to perform retrieval, they fall short to lexical models due to their coarseness and lack of specificity. In particular, Dumais [68] showed that LSI retrieves irrelevant documents in high positions of the ranking list due to its lack of specificity. Blei et al. [28] found that pLSI does not provide a probabilistic model at the document level. In fact, each document is represented as a list of mixing proportions for topics, but there is no generative probabilistic model for these numbers. This leads to two critical factors: 1) there is no inherent way to predict a previously unseen document, and 2) the number of parameters grows linearly with the number of training documents – making pLSI susceptible to overfitting. Regarding LDA, Wei and Croft [238] reported that relying solely on LDA-based representations to perform retrieval hurts performances. Given that LDA represents documents and queries as random mixtures over latent topics, its representations might not be as precise as bag-of-words representations and be too coarse to be used alone for retrieval.

3.4.2 Neural IR Models

In the last few years, the increased availability of data and the success of (deep) neural networks in the NLP field have promoted the diffusion of neural models in the IR field. Similarly to Onal et al. [173], we classify existing neural IR approaches into two categories: representation-based and interaction-based. Representation-based approaches learn (or exploit) distributed representations and then use them to perform semantic matching. Depending on the task and model considered, the learned representations can be used to perform semantic similarity between a pair of words (e.g., to identify expansion terms for query expansion) or semantic matching between a query and a document (e.g., to perform document retrieval). On the other hand, interaction-based approaches first build joint representations of query/document pairs and then pass them through a neural network, which predicts the relevance score of query/document pairs. In plain words, interaction-based models predict the relevance scores of query/document pairs by computing the interactions between query and document terms. All the models within this category are end-to-end supervised approaches that employ (deep) neural networks to perform re-ranking over an initial set of candidate documents, previously retrieved by a lexical (efficient) model.

In the following, we review both representation- and interaction-based approaches. However, we are more interested in models that meet two requirements: (i) they can be applied regardless of the presence of labeled data, and (ii) they can be used at the early stages of the IR pipeline. Given that the objective of this thesis is to address the semantic gap between queries and documents, models that do not require labeled data allow us to investigate domains with a high social value, where the semantic gap is prominent, and explicit relevance labels are scarce and expensive resources. On the other hand, relying on models that can be used at the early stages of the IR pipeline is fundamental to address the semantic gap – otherwise, relevant documents most affected by this gap will remain undiscovered. Therefore, we focus our attention on unsupervised representation-based neural IR models. Unsupervised representation-based neural IR models can be applied to any domain, as they do not require any labeled data for training, and are often used to perform retrieval over the entire document collection or in combination with lexical models for query expansion or rank fusion strategies. In other words, they are typically used at the early stages of the IR pipeline rather than in re-ranking scenarios.

Representation-Based Neural IR Models

The advances in representation learning have led the IR community to develop retrieval models based on distributed representations of words and documents. Regarding unsupervised retrieval models, we can divide them into two groups: (A) approaches incorporating features from representation learning models, and (B) approaches learning representations of words and documents from scratch.

Within (A), Vulić and Moens [235] proposed to compose document representations as the weighted sum of their word embeddings. The method uses the self-information [46] value of each word as its weighting operator. The idea is that corpus-based weights, like Inverse Document Frequency (IDF) or self-information, assign more importance to words bearing more information content during the compositional process. Zuccon et al. [257] combined a traditional retrieval model with a translation model that uses word embeddings to estimate probabilities. Similarly, Ganguly et al. [86] presented a generalized language model where the mutual independence between a pair of words no longer holds and word embeddings are used to derive the transformation probabilities between words. Guo et al. [96] introduced the Bag-of-Word-Embeddings (BoWE) model. BoWE represents every document as a matrix of its word embeddings and then models the matching between queries and documents as a non-linear word transportation problem. Ai et al. [11] evaluated the effectiveness of doc2vec DBOW for ad hoc retrieval and – inspired by Levy and Goldberg [139] – performed a deeper analysis later in [10]. The analysis formally addressed three intrinsic problems of the DBOW architecture that limit its effectiveness in retrieval tasks: the susceptibility to short documents overfitting, the excessive suppression of frequent words importance, and the lack of word-context information. Regarding embedding-based query expansion methods, Zamani and Croft [251, 252] proposed to use pre-trained word embeddings to incorporate and weight terms that do not occur in the query – but are semantically related to the query terms – or as an embedding-based relevance model for PRF. Along the same lines, Kuzi et al. [136] presented a suite of query expansion methods based on word2vec CBOW embeddings. The expansion terms identified by the embedding-based methods were either used to expand the original query or integrated in PRF based methods. Diaz et al. [63] investigated the effectiveness of local embeddings – learned on topically-constrained corpora – compared to global embeddings – learned on large topically-unconstrained corpora – for query expansion.

The approaches in (A) proved to be effective thanks to their combination with lexical retrieval models. Besides, the presented approaches are general and can be applied to any model that provides the required representations (i.e., words, concepts, or documents). Therefore, most of these approaches can be applied to the novel unsupervised knowledge-enhanced neural framework we present in Chapter 6 – in particular embedding-based query expansion methods [136, 251, 252]. However, the focus of Chapter 6 is the integration of external knowledge in neural retrieval models to address the semantic gap. Thus, we do not apply the reviewed approaches to our framework, but rather we investigate its effectiveness compared to other (knowledge-enhanced) neural IR models in terms of relevant documents retrieved at the early stages of the IR pipeline. Naturally, developing distributed representations of words and documents that are better suited for IR tasks has a positive effect also on the reviewed approaches.

Within (B), Van Gysel et al. [224] introduced an end-to-end representation learning model for expert search that outperforms statistical vector space models [58] and generative language models [20, 21]. The proposed model employs only textual evidence to learn word representations – thus avoiding explicit feature engineering – for retrieving experts in online document collections. Van Gysel et al. [222] presented the Latent Semantic Entities (LSE) model, a vector space model that jointly learns the representation of words, e-commerce products, and the mapping between them without explicit annotations. LSE directly models the discriminative relation between products and a particular word. The experimental results showed that LSE constructs better product representations than LSI [57], LDA [28], and word2vec [162] models. Then, Van Gysel et al. [223] presented the Neural Vector Space Model (NVSM), which learns word and document representations from scratch without considering any external source of information. NVSM extends previous models [222, 224] in three ways: increasing regularization, reducing the internal covariate shift, and incorporating term specificity within word representations. The results showed that NVSM significantly outperforms LSE in newswire retrieval. Given the prominence of NVSM within unsupervised neural IR models, we perform an in-depth analysis of NVSM and the semantic signals it provides throughout Chapter 4. In particular, we describe NVSM and reproduce the experiments performed by Van Gysel et al. [223] in Section 4.3.

The contribution of our neural framework over (B) lies in the integration of external knowledge within neural vector space models to bridge the semantic gap between queries and documents. Compared to NVSM [223], the proposed framework jointly learns word, concept, and document representations. The learned representations are optimized for IR and encode linguistic features that are crucial to address the semantic gap between queries and documents. Similarly to NVSM, the framework does not require any labeled data for training and can be applied to any domain where external knowledge resources are available.

Regarding supervised retrieval models, one of the earliest approaches is the Deep Structured Semantic Model (DSSM) [116]. DSSM is an end-to-end supervised approach that takes as input a query/document pair and passes it through a deep neural network. The neural network consists of three non-linear layers stacked on top of a word hashing layer. The query and the document are first modeled as bag-of-words representations. Then, each word is mapped to a vector of character tri-grams by the word hashing layer to cope with large-scale collections and vocabularies. For instance, the word "fruit" is mapped to [#fr, fru, rui, uit, it#], where the # symbol refers to the start and end of the word. The (low-dimensional) tri-gram vector serves as input to the non-linear layers of the network. Other than reducing the vocabulary size significantly, tri-gram hashing also helps to address out-of-vocabulary words not seen during training. During training, DSSM learns to maximize the conditional likelihood of clicked documents given a query using click-through data. In this sense, DSSM is one of the first models incorporating click-through data in deep neural networks. However, DSSM requires large-scale training data for its huge parameter size and it is typically used to perform re-ranking due to efficiency limitations [95].

DSSM inspired other researchers to propose architectural variants or novel ways of using its distributed representations. Regarding architectural variants, the Convolutional Latent Semantic Model (CLSM) [201] replaces the Feed Forward Neural Network (FFNN) of DSSM with Convolutional Neural Networks (CNNs), whereas the LSTM Deep Structured Semantic Model (LSTM-DSSM) [175, 176] with LSTM networks. On the other hand, Li et al. [141] exploited the distributed representations learned by DSSM and CLSM to re-rank documents based on in-session contextual information, whereas Ye et al. [248] generalized DSSM variants by distinguishing the clicked query/document pairs with different relevance information – both semantic and lexical.

Interaction-Based Neural IR Models

Unlike representation-based deep matching models such as DSSM [116], interaction-based models address the problem of predicting the relevance score of a query/document pair by computing the interactions between the query and document terms. Pang et al. [177] were the first to apply interaction-based deep matching models to ad hoc retrieval. The authors evaluated two interaction-based deep matching models: Convolutional Matching Model Architecture-II (ARC-II) [115] and MatchPyramid [178]. ARC-II builds local interactions between query/document pairs by summing up word embeddings in a small (sliding) context window and then employs convolutional layers to extract sequential and hierarchical features from these interactions. Similarly, MatchPyramid first builds a matching matrix that represents the local interactions computed between query and document embedding-based representations. Then, it passes the built matching matrix through a CNN to learn hierarchical matching patterns. The learned high-level matching patterns are finally fed to a Multi Layer Perceptron (MLP) to produce the matching score of query/document pairs. The results obtained by Pang et al. [177] showed that MatchPyramid significantly outperforms several representation- and interaction-based deep matching models [116, 115], but fails when compared to lexical models like BM25 [192] and Query Likelihood Model (QLM) [254].

The first model to show improvements over lexical models is the Deep Relevance Matching Model (DRMM) [95]. DRMM relies on semantic and lexical matching signals to compute the local interactions between pairs of query/document terms. For each query term, DRMM maps the variable-length local interactions into a fixed-length matching histogram. The matching histograms are then fed into a FFNN to learn hierarchical matching patterns and produce a matching score for each query term. Finally, a term gating network computes the aggregation weights that are used to combine the scores from each query term and produce the overall matching score for the query. The network is trained using a margin ranking loss function. The experimental results performed by Guo et al. [95] showed that DRMM outperforms other representation- and interaction-based deep matching models, such as DSSM [116], ARC-II [115], and MatchPyramid [178, 177]. Thus, given the impact that DRMM had on deep matching models for IR – and in general on neural IR – we perform an in-depth analysis throughout Chapter 4, where we compare DRMM with other lexical and semantic models to understand what features lexical and semantic models share and if their signals are complementary. In particular, we provide a detailed description of DRMM and its training process in Section 4.2, where we also reproduce the experiments performed by Guo et al. [95].

Other successful approaches in this category are Match-SRNN [236], Hierarchical Neural maTching model (HiNT) [72], Kernel-based Neural Ranking Model (K-NRM) [239], Convolutional Kernel-based Neural Ranking Model (Conv-KNRM) [53], Position-Aware Convolutional-Recurrent Relevance Matching (PACRR) [117], and Context-aware PACRR (Co-PACRR) [118]. Match-SRNN [236] models the interaction between two texts as a recursive process. This means that the interaction of two texts at each position can be considered as a combination of the interactions between their prefixes as well as the word-level interaction at the current position. Match-SRNN adopts an approach similar to MatchPyramid [178], but replaces the CNN with a Spatial Recurrent Neural Network (SRNN). HiNT [72] focuses on the different relevance patterns that are present in a document given a query. The underlying assumption is that a document can be completely (or partially) relevant to a query as long as it provides sufficient information to the user need(s). Hence, HiNT allows relevance signals at different granularities to compete with each other for the final relevance assessment through a hierarchy of matching layers. K-NRM [239] relies on a translation matrix to model word-level similarities via word embeddings, a kernel-pooling technique to extract multi-level soft match features, and a learning-to-rank layer to combine these features into the final ranking score. Conv-KNRM [53] extends K-NRM by adding soft matching between word n-grams using CNNs. PACRR [117] computes the relevance score of query/document pairs from multiple word n-gram similarity matrices processed first with a CNN and then with a Recurrent Neural Network (RNN). Finally, Co-PACRR [118] extends PACRR in two ways: by employing a context vector to enrich the matching signals, and by replacing the RNN with a simpler FFNN.

3.5 Knowledge-Enhanced Semantic Models

As shown in Sections 3.2 and 3.4, two main lines of work have emerged in the past years to address the semantic gap between queries and documents: (i) the use of external knowledge resources to enhance bag-of-words representations in lexical models, and (ii) the use of semantic models to perform semantic matching between latent representations of queries and documents. However, even though semantic models based on the distributional hypothesis [102] capture latent relationships between textual units relying only on the document collection, they are hampered by two main limitations. First, these models fail to discriminate polysemous words, as they learn unique representations for words regardless of the context in which these words occur. Secondly, these models fail to learn close representations for synonyms occurring in different contexts, as they lack the relational knowledge required to identify synonymy relationships between words.

Hence, the integration of external knowledge in the learning process of semantic models can further improve their effectiveness towards the semantic gap. In this section, we review those approaches that integrate external knowledge in the learning process of traditional semantic models (see Subsection 3.4.1) and neural IR models (see Subsection 3.4.2).

3.5.1 Knowledge-Enhanced Traditional Semantic Models

Regarding the integration of external knowledge in the leaning process of traditional semantic models, Guo et al. [94] proposed the Knowledge-Enhanced LSI (KELSI). KELSI extends LSI [57] in two ways: (i) by augmenting the original term-by-document matrix with additional concept-based vectors constructed from external knowledge resources, and (ii) by applying different query reformulation methods – that exploit external knowledge resources – during semantic matching. The experimental results, conducted on the OHSUMED collection [107] and relying on MeSH [146] and UMLS [29] thesauri as knowledge resources, showed that KELSI provides significant performance gains over LSI. To the best of our knowledge, this is the only approach that has been developed to enhance traditional semantic models with external knowledge.

3.5.2 Knowledge-Enhanced Neural IR Models

Motivated by the recent advancements of the NLP community in the integration of external knowledge within representation learning models (see Subsection 3.3.2), IR researchers have started to develop knowledge-enhanced neural IR models. However, being knowledge-enhanced representation models quite recent, there are only few approaches proposed for

IR tasks. Besides, given the objective of this thesis – and the model requirements we are interested in – we only review unsupervised representation-based neural IR models integrating external knowledge in the learning process of word and document representations.

Liu et al. [147] exploited word relations from a medical knowledge resource to constrain the word representations learned by word2vec. The underlying idea is that related words within the knowledge resource should have similar representations. The constrained word representations were then used to perform document re-ranking. The results showed that constrained word representations are more effective than corpus-driven word representations when used together with bag-of-words models for re-ranking. Nguyen et al. [169] presented two models: the conceptual doc2vec (cdoc2vec) and the retrofitted doc2vec (rdoc2vec). Similar to the model proposed by Devine et al. [56], cdoc2vec learns document representations built upon concepts that have been previously extracted from text. Then, rdoc2vec retrofits document representations by minimizing the distance between doc2vec and cdoc2vec representations. The learned representations were injected in a text-to-text matching process according to a PRF based query expansion strategy. Nguyen et al. [170] proposed a tri-partite neural language model that leverages explicit knowledge to jointly constrain word, concept, and document representations. The authors applied the model in two IR tasks: document re-ranking and query expansion. Tamine et al. [220] extended [169, 170] to investigate the combined use of corpus-based information and external knowledge resources in different NLP and IR tasks. The authors compared the impact of the different learning approaches on the quality of the learned representations. They found that rdoc2vec and tri-partite models show the same level of performance in identifying relevance signals for IR tasks.

To better understand knowledge-enhanced semantic models and their effectiveness for IR tasks, we perform a reproducibility study of the seminal works by Liu et al. [147] and Nguyen et al. [169] in Chapter 6. Specifically, we describe the knowledge-enhanced word embeddings used [250, 74], and proposed [147], by Liu et al. in Section 6.2, and the knowledge-enhanced document embeddings proposed [169] by Nguyen et al. in Section 6.3. Then, we present our unsupervised knowledge-enhanced neural framework in Section 6.5. Compared to the reviewed models, our framework shows similarities with the works of Liu et al. [147] and Tamine et al. [220]. In particular, the framework constrains synonym representations similarly to Liu et al. [147] and learns word, concept, and document representations as in Tamine et al. [220]. Nevertheless, the framework models polysemy by combining word and concept representations in the learning process. This creates contextual representations that the model of Liu et al. [147] and those of Tamine et al. [220] do not handle. Furthermore, an important difference between the works of Liu et al. [147], Tamine et al. [220], and our work is that we optimize the framework for IR. Conversely, the models proposed by Liu et al. [147] and

Tamine et al. [220] are extensions of neural language models – which are optimized for NLP tasks. Therefore, (knowledge-enhanced) neural language models do not encode relevance signals or discriminative aspects between queries and documents – which are fundamental to effectively address IR tasks. This difference reflects on the different loss functions used to train the framework and the knowledge-enhanced neural language models. To the best of our knowledge, the framework we present is the first unsupervised knowledge-enhanced framework that learns word, concept, and document representations specifically for IR.

Chapter 4

Lexical and Semantic Signals

Traditionally, IR models rely on lexical matching signals to perform retrieval [211, 192, 254, 13]. Given query and document bag-of-words representations, lexical models compute the relevance score using heuristics defined over the lexical overlap between query and document representations. Although successful, these models struggle to address the semantic gap between queries and documents. For instance, when a query and a document use different words to express the same concept, lexical models fail to match them. To bridge this gap, semantic models have been used for decades in IR [57, 113, 28]. Semantic models rely on semantic matching signals between query and document latent representations to perform retrieval. However, traditional semantic models such as LSI [57], pLSI [113], and LDA [28] fall short to lexical models due to their coarseness and lack of specificity [68, 238].

Recently, the advances and the success of deep learning in many different tasks have promoted the diffusion of (deep) neural networks in IR, reviving the interest in semantic models in the research community. Since the very first approaches [116, 210], neural IR has attracted a lot of attention: dedicated workshops were held at the ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR) in 2016 [49] and 2017 [47], while an in-depth monograph [165] and a special issue in the Information Retrieval Journal (IRJ) [48] were published in 2018. Also, SIGIR papers employing deep learning are increasing at a fast pace – i.e., from two articles published in 2014 to eleven articles published in 2017 [8] to more than fifty articles published in 2020.

Nevertheless, the research community has also raised concerns about the actual efficiency and effectiveness of neural IR models [144]. To this end, Wei et al. [245] critically examined the advances in neural IR regarding the test collections used, the improvements over robust and well-tuned baselines, and the reproducibility of the methods. Thus, along with the growing importance of neural IR models, their reproducibility is becoming a central topic too. Anyway, the issue of reproducibility of retrieval systems regards the IR field as a whole, not the only neural IR. Reproducibility efforts focus on several core topics in IR, ranging from reproducing baselines [145, 243] and core IR components [202] to evaluation [82, 114] and advanced applications [103]. Reproducibility is now a core research topic in IR, with dedicated workshops [76, 14, 42], a specific track at the European Conference on Information Retrieval (ECIR) since 2015, and dedicated journal special issues [77, 78].

A neural IR model is an ecosystem of components and its reproducibility, even when the source code is available, is quite challenging. Indeed, neural IR models often include text processing techniques, bag-of-words representations, word embeddings, optimizers, query expansion techniques, and other traditional IR and NLP components – which are used to feed a shallow or deep neural network. Every single component has a sizable impact on the performance of the model, that is often overlooked. For instance, how documents are preprocessed has implications on the learning of word embeddings. In turn, word embeddings affect the optimizer and parameter selection, and so on.

Even though this domino effect holds true for almost all advanced IR systems, it is particularly accentuated for neural models – where it is hard to understand the rationale behind a specific output and to detect a component not working correctly. To reproduce the results achieved by neural models, each component needs to be properly tuned. Besides, describing a neural network architecture in detail or providing the source code is usually not sufficient to reproduce the model successfully. Furthermore, generalizing neural IR models to different collections than the tested ones accentuates the problem even more, as we need to adapt and optimize many components to different settings – be them a new domain, language, or task. Therefore, the analysis of neural models, especially through reproducibility studies [69, 247, 79, 80], becomes crucial for in-depth understanding. The more we understand the single components and their interactions, the more we can generalize the approach and successfully transfer it to different domains and tasks.

In this chapter, we investigate lexical and semantic matching signals. We want to understand what features lexical and semantic models share, if their signals are complementary, and how they can be combined to effectively address the semantic gap. In particular, we evaluate the critical aspects of neural IR models through different analyses. Each analysis brings a different perspective in the understanding of semantic models and their relation with lexical models. To this end, we reproduce, evaluate, and generalize two neural models: DRMM and NVSM. DRMM [95] is an interaction-based deep matching model that exploits semantic and lexical matching signals to compute the local interactions between pairs of query/document terms. DRMM achieves competitive results in re-ranking and it is still one of the reference neural IR approaches. NVSM [223] is an unsupervised representation-based neural model that jointly learns word and document representations to perform semantic matching between documents and queries. NVSM yields effective results in document retrieval on TREC newswire collections. Besides, NVSM is one of the very few existing end-to-end unsupervised neural IR models. Therefore, it has great potential to generalize, as it does not require any interaction or labeled data – which are scarce and expensive resources in a typical IR experimental setting.

We reproduce the experiments performed by Guo et al. [95] with DRMM relying on the source code shared by the authors. On the other hand, we re-implement NVSM from scratch in Python using TensorFlow – a widely-used and consolidated library for deep learning.[1] This choice enables a straightforward comparison of NVSM with many other neural IR models available in public repositories.[2] We reproduce the experiments performed by Van Gysel et al. [223] not only for NVSM but also for the main baselines they considered.

Then, we consider four different perspectives in the analysis of DRMM and NVSM. First, we perform an in-depth evaluation of DRMM and NVSM, where we compare them with well-known lexical models, that is TF-IDF [196], BM25 [192], QLM [254], and Divergence From Randomness (DFR) [13], and with semantic models, such as word2vec based models [235]. In this way, we investigate the potential and limitations of semantic models compared to lexical models. Understanding neural IR strengths and weaknesses can enhance the integration of neural models into multi-stage IR systems – which employ a variety of pre- and post-retrieval components, such as query expansion and relevance feedback. Secondly, we generalize DRMM and NVSM to different scenarios for which they were not initially designed: (i) Web, where we consider the TREC WT2g collection [104]; (ii) medicine, where we consider the OHSUMED collection [107]; (iii) multilingualism, where we consider Italian, German, and Farsi [60, 3] newswire collections from CLEF. In other words, we evaluate how different domains and languages affect their performances. Note that to avoid incurring in memory (and time) issues, we chose collections of the same order of magnitude as those adopted by Van Gysel et al. for NVSM [223]. Thirdly, we perform an analysis of the impact of word representations learned by different models on DRMM. The models considered are: word2vec [163], fasttext [30], and NVSM [223]. Given that neural language models, such as word2vec or fasttext, learn different word representations and produce different matching scores for the same terms [182], we want to understand to what extent they impact the ability of DRMM to learn query/document interactions. Finally, we perform a topic-by-topic analysis and comparison between neural models and BM25. We highlight the differences in performance among models – describing the topics where neural models perform better than lexical ones, and vice versa. That is, we investigate whether lexical and

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

[2] https://github.com/NTMC-Community/MatchZoo; https://github.com/Georgetown-IR-Lab/OpenNIR

semantic matching signals are complementary and to what extent lexical and neural models retrieve different relevant documents.

The main contributions of this chapter are:

C1 We perform a reproducibility study of DRMM and NVSM on the original test collections considered by Guo et al. [95] and Van Gysel et al. [223], respectively.

C2 We compare DRMM and NVSM with well-known lexical and semantic models.

C3 We evaluate the impact that different domains and languages have on DRMM and NVSM. That is, we generalize DRMM and NVSM to different domains and tasks.

C4 We evaluate the impact that different word embeddings have on DRMM performances.

C5 We perform a topic-by-topic evaluation between DRMM, NVSM, and BM25 to identify similarities and differences between lexical and semantic matching signals.

The rest of this chapter is organized as follows. We describe the setup employed in our experiments in Section 4.1. We describe and reproduce DRMM and NVSM in Sections 4.2 and 4.3, respectively. We compare DRMM and NVSM with well-known lexical and semantic models in Section 4.4, whereas we test the robustness of DRMM and NVSM to different domains and languages in Section 4.5. We assess the impact of different word embeddings on DRMM in Section 4.6. We perform an in-depth topic-by-topic analysis between lexical and semantic matching signals in Section 4.7. Finally, Section 4.8 concludes the chapter with a discussion on the lessons learned.

4.1 Experimental setup

We use eleven test collections in different languages and from various domains to evaluate DRMM and NVSM. The statistics for the test collections originally used to evaluate DRMM and NVSM are shown in Table 4.1, whereas the statistics for the test collections used to evaluate how well DRMM and NVSM generalize to different scenarios are shown in Table 4.2.

We consider three variants of the query fields: *title* (t), *description* (d), and *title+description* (t+d). We compare DRMM and NVSM with well-known lexical and semantic models used in IR. The lexical models are TF-IDF [196], BM25 [192], QLM [254], and the Poisson estimation for randomness using Laplace succession for normalisation (PL2) model [13], typically known as DFR. On the other hand, the semantic models include word2vec [163] and LDA [28]. For QLM, we employ the approaches that rely on Jelinek-Mercer smoothing,

Table 4.1 Statistics of the AP88-89, FT, LA, WSJ, Robust04 and NY collections. Query count does not consider topics for which relevance judgments are not available.

	AP88-89	FT	LA	NY	Robust04	WSJ
Vocabulary	247,725	437,511	197,024	1,062,137	760,467	184,717
Document Count	164,597	210,158	131,896	1,855,658	528,155	173,252
Query Count	149	144	143	50	249	150

Table 4.2 Statistics of the WT2g, OHSUMED, CLEF-IT, CLEF-DE, and CLEF-FA collections. Query count does not consider topics for which relevance judgments are not available.

	WT2g	OHSUMED	CLEF-IT	CLEF-DE	CLEF-FA
Vocabulary	1,049,056	265,923	232,335	739,053	399,185
Document Count	247,491	348,566	157,558	223,132	166,774
Query Count	50	97	90	95	100

referred to as QLM (jm), and Dirichlet smoothing, referred to as QLM (dir) [254]. For word2vec, we employ the approach originally proposed by Vulić and Moens [235], who define document representations as the weighted sum of their word embeddings. We consider the unweighted sum, referred to as word2vec (add), and the sum weighted by terms self-information values, referred to as word2vec (si) – where self-information is a term specificity measure similar to IDF [46].

4.1.1 Reproducibility Study

DRMM

For this reproducibility study, we reproduce the experiments performed by Guo et al. with DRMM on the Robust04 collection [232]. We consider the complete set of 250 topics and we evaluate DRMM considering the *title* (t) and the *description* (d) fields. We consider the same evaluation measures adopted by Guo et al. [95], which are MAP, nDCG@20, and P@20.

NVSM

For this reproducibility study, we reproduce the experiments performed by Van Gysel et al. [223] with NVSM on six TREC newswire collections. Four of these six collections are subsets of the TIPSTER corpus [99], namely AP88-89, FT, LA, and WSJ [101]. The remaining two collections are Robust04 [232] and NY [12]. In all collections, we consider the *title* (t) field of the given topics. We employ the same measures adopted by Van Gysel et al. [223] to evaluate NVSM, which are MAP, nDCG@100, and P@10. Following the authors' work, we also perform a two-tailed paired Student's t-test between word2vec (si) and NVSM to test statistical significance.

The reader can find more details on the test collections used for the reproducibility study in Subsection 2.1.4.

4.1.2 Comparison between Lexical and Semantic Models

We compare lexical and semantic models on Robust04 and NY collections. For each collection, we consider the *title* (t) field of the given topics. The lexical models considered are TF-IDF, BM25, QLM, and DFR. All the models perform stemming using the Krovetz stemmer [133]. On the other hand, the semantic models considered are word2vec (add), word2vec (si), DRMM, and NVSM. The objective is to evaluate the performances of lexical and semantic models, also compared to state-of-the-art IR approaches. To this end, we report, when available, the performances of state-of-the-art approaches, such as BM25+RM3 [244], BERT applications for ad hoc retrieval [246], and best systems from TREC.

We use MAP, nDCG@100, and P@10 to evaluate models. We perform the post-hoc Tukey's Honest Significant Differences (HSD) test [221] with one-way ANOVA to test statistical significance. The Tukey's HSD test checks all pairwise differences between runs and, as indicated in [37, 84], it is a viable method for dealing with the multiple comparisons problem [221]. We apply the Tague-Sutcliffe transformation to Tukey's HSD tests [218]. This transformation is applied only if either Lilliefors or the Jarque-Bera test rejects the normality hypothesis for any experiment.

4.1.3 Collection-based Evaluation

We evaluate the impact that different domains and languages have on the performances of DRMM and NVSM. To this end, we also compare DRMM and NVSM to lexical models. The lexical models considered are TF-IDF, BM25, QLM, and DFR. The collections used to perform this evaluation are: WT2g [104], OHSUMED [107], CLEF-IT, CLEF-DE, and CLEF-FA [60, 3]. For OHSUMED, we consider all the 106 topics. Then, for each collection, we choose the topic combinations where DRMM and NVSM perform best – which are also the most common. The considered topic combinations are: WT2g (t+d), OHSUMED (d), CLEF-IT (t+d), CLEF-DE (t+d), and CLEF-FA (t+d). For CLEF collections, we rely on publicly available stoplists that are specific to the target language.[3] We use MAP, nDCG@100, and P@10 to evaluate models. Then, we employ the post-hoc Tukey's HSD test to assess

[3] http://members.unine.ch/jacques.savoy/clef/

statistical significance. Whenever possible, we report the performances of the best TREC and CLEF systems.

The reader can find more details on the test collections used for the collection-based evaluation in Subsections 2.1.1 and 2.1.4.

4.1.4 Embedding-based Evaluation

We evaluate how word embeddings – learned by different models – affect the performance of DRMM. Using different word representations can have a sizable impact on the performance of neural models – as also shown by MacAvaney et al. [153], who investigate how BERT [59] and ELMo [180] representations can be used by neural IR models.

We consider four different types of word embeddings:

word2vec (corpus): word2vec embeddings learned on the considered test collections. We rely on Gensim [183] and we follow the instructions provided by Guo et al. [95]. As preprocessing, we remove punctuation and stopwords relying on the INQUERY stoplist [36], we perform stemming using Krovetz stemmer, and we remove all the terms that contain digits and are shorter than three characters. For training, we set the context window size to 10, the embedding size to 300, and the number of negative samples to 10. Furthermore, we subsample terms whose frequency is greater than 10^{-4} and we discard terms appearing less than 10 times in the collection. We train word2vec for 10 epochs – we empirically selected 10 as the number of training epochs after evaluating DRMM performances using word2vec embeddings from a model trained for 5, 10, 15, and 20 epochs.

word2vec (Google): word2vec embeddings trained by Google on part of the Google News dataset (about 100 billion words). word2vec (Google) contains 300-dimensional vectors for 3 million words and phrases.[4]

fasttext: fasttext embeddings learned with Gensim on the considered test collections. We follow the instructions provided by Guo et al. [95] also for fasttext. In particular, we set the context window size to 10, the embedding size to 300, and the number of negative samples to 10. Again, we subsample terms whose frequency is greater than 10^{-4} and we discard terms appearing less than 10 times in the collection. As with word2vec (corpus), we train fasttext for 10 epochs on each of the considered test collections.

[4] The word2vec (Google) model is available at: https://code.google.com/archive/p/word2vec/

NVSM: word embeddings learned by NVSM. NVSM jointly learns word and document representations from scratch without the need for annotated data. The learned representations are then used to perform retrieval. Details on NVSM training and retrieval processes are provided in Section 4.3. Since NVSM learns word representations as word2vec, we can feed its word embeddings to DRMM without any modification. As done for word2vec (corpus) and fasttext, we train NVSM on each of the considered test collections.

The test collections considered are Robust04, NY, WT2g, and OHSUMED. The topic combinations used are: *title* (t) for Robust04 and NY, *title+description* (t+d) for WT2g, and *description* (d) for OHSUMED. As evaluation measures, we adopt MAP, nDCG@100, and P@10.

4.1.5 Topic-based Evaluation

We conduct a topic-by-topic analysis and comparison between DRMM, NVSM, and BM25. We want to understand the differences in performance between these models, in what topics they succeed/fail, and why. To perform this evaluation, we consider Robust04, NY, WT2g, OHSUMED, and CLEF collections. The topic combinations used are: *title* (t) for Robust04 and NY, *title+description* (t+d) for WT2g, *description* (d) for OHSUMED, and *title+description* (t+d) for CLEF collections.

Additionally, we employ Kernel Density Estimation (KDE) [237] to estimate the Probability Density Function (PDF) of the AP@1000 of BM25, DRMM, and NVSM for all the topics. Then, we compute the Kullback-Leibler Divergence (KLD) [134] between these PDFs to get an estimate of the difference in AP@1000 distributions for each model.

4.2 Reproducibility Study: DRMM

We describe and reproduce the Deep Relevance Matching Model (DRMM), an interaction-based deep matching model for ad hoc retrieval proposed by Guo et al. [95]. As shown in Subsection 3.4.2, interaction-based models are supervised end-to-end deep neural networks that build joint representations of query/document pairs to predict the relevance score between queries and documents. DRMM is a prominent example of this category of models, and it is still one of the reference neural IR approaches.

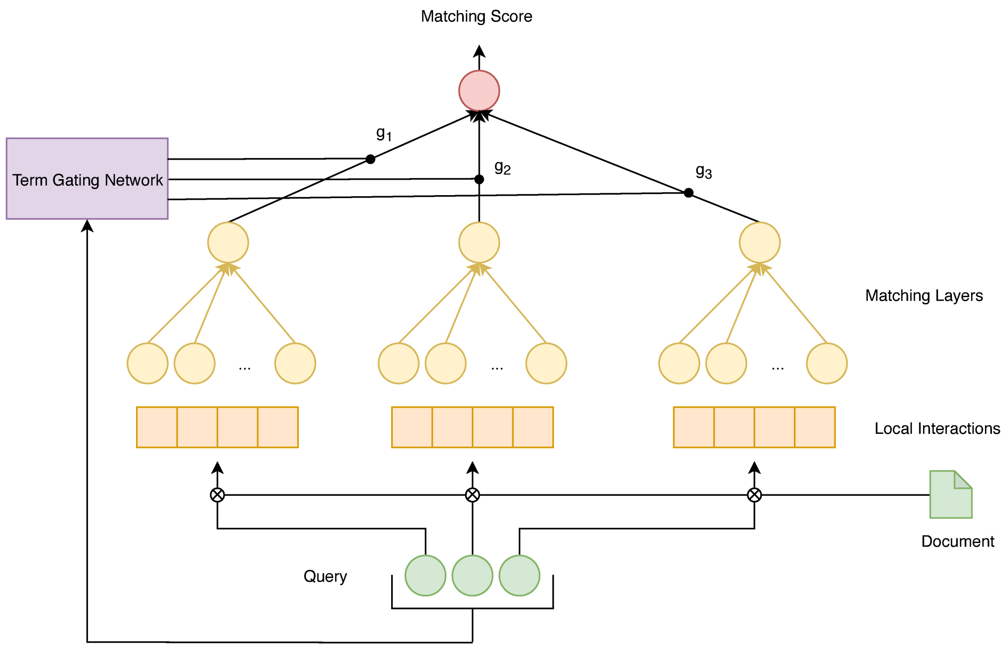


Fig. 4.1 Deep Relevance Matching Model (DRMM) neural architecture.

4.2.1 The Deep Relevance Matching Model

DRMM employs a joint deep architecture at the query term level for relevance matching, implementing the following strategies: (i) it considers semantic and exact – i.e., lexical – matching signals between query and document terms; (ii) it enables different importance weights for query terms; and (iii) it complies with different matching requirements, based on the verbosity and scope hypotheses. The verbosity hypothesis assumes that a long document is like a short document, covering a similar scope with more words. Vice versa, the scope hypothesis assumes that, in longer documents, only portions of the content may be relevant and the whole document is therefore not required to be relevant to a query.

Based on semantic and exact matching signals, DRMM considers local interactions between each pair of query/document terms. For each query term, DRMM maps the variable-length local interactions into a fixed-length matching histogram. Then, these matching histograms are fed into a FFNN that learns hierarchical matching patterns and outputs a matching score for each query term. Finally, a term gating network computes the aggregation weights that are used to combine the scores from each query term and produce the overall matching score for the query. DRMM neural architecture is depicted in Figure 4.1.

The local interactions between each pair of terms from a query and a document are obtained computing the cosine similarity between term embeddings. Scores are then aggregated in matching histograms, which discretize the interval $[-1,1]$ into a set of k bins. For instance, if we take 0.5 as the bin size, each bin will contain the cosine similarity scores from the intervals: $[-1,-0.5)$, $[-0.5,0)$, $[0,0.5)$, $[0.5,1)$. Then, exact matching is treated as a separate $k+1$ bin. In this way, the matching histogram is able to distinguish the semantic signals from the lexical ones.

Guo et al. proposed three different ways to map the values in the matching histograms:

Count-based Histogram (CH): it considers the count of local interactions in each bin as the histogram value.

Normalized Histogram (NH): it normalizes the count value in each bin by the total count.

LogCount-based Histogram (LCH): it applies logarithm over the count value in each bin.

The number of bins considered by Guo et al. was 30, and the matching histograms configuration that performed best was LCH. Therefore, in our experiments we set the number of bins to 30 and we rely on the LCH configuration.

The FFNN consists of two layers, the first layer presents five nodes, whereas the second layer only one. Both layers adopt $\tanh(\cdot)$ as their activation function. The matching scores returned by the FFNN are then weighted with coefficients computed by a term gating network. The term gating network produces an aggregation weight for each query term controlling how much the relevance score on that query term contributes to the final relevance score:

$$g_i = \frac{\exp(\boldsymbol{w}_g \boldsymbol{x}_i^{(q)})}{\sum_{j=1}^{M} \exp(\boldsymbol{w}_g \boldsymbol{x}_j^{(q)})}, \quad i = 1, ..., M \tag{4.1}$$

where $\boldsymbol{w}_g$ is the weight vector of the term network and $\boldsymbol{x}_i^{(q)}$, $i = 1, ..., M$ denotes the i-th query term. Guo et al. developed different types of weighting functions, which require different input values:

Term Vector (TV): In this method, $\boldsymbol{x}_i^{(q)}$ denotes the i-th query term vector, and $\boldsymbol{w}_g$ is a weight vector of the same size of the term vectors.

Inverse Document Frequency (IDF): In this method, $\boldsymbol{x}_i^{(q)}$ denotes the IDF of the i-th query term, and $\boldsymbol{w}_g$ is a coefficient with a single parameter.

Guo et al. found out that the best performing approach to compute these weights is the IDF weighting scheme. Therefore, we consider only the IDF weighting scheme in our experiments.

Finally, DRMM is trained using the hinge loss:

$$L(\Theta | q, d^+, d^-) = \max(0, 1 - s(q, d^+) + s(q, d^-)) \tag{4.2}$$

where Θ includes the feed-forward and term gating network parameters, d^+ is a document ranked higher than d^- for a given query q, and $s(\cdot)$ is the function that computes the matching scores.

4.2.2 Implementation Details

The input data and the implementation details of DRMM are publicly available at: https://github.com/faneshion/DRMM/. Nevertheless, to reproduce the original results and to generalize DRMM to different domains and languages, we had to define a new preprocessing pipeline and to develop a new training script for the word embeddings required by the model. We adopt the DRMM configuration that performs best in the experiments performed by Guo et al. [95], that is we consider LCH matching histograms with 30 bins, IDF scores in the term gating network, and we set the feed-forward first layer size to 5.

DRMM adopts a re-ranking strategy for efficient computation. Therefore, the model re-ranks an initial set of candidate documents, previously retrieved by a lexical (efficient) model. Guo et al. [95] relied on QLM with Dirichlet smoothing [254] to retrieve an initial set of 2000 candidate documents. Besides, during indexing and retrieval, they removed stopwords using the INQUERY stoplist [36] and stemmed words using the Krovetz stemmer [133]. We apply the same preprocessing but we implement QLM using a modified version of Terrier 4.1 [154],[5] rather than Galago.[6] We decided to rely on a different library to implement QLM to evaluate whether different implementations of the same model, under the same conditions, provide comparable results. We keep this setup for all the experiments we perform with DRMM.

The most critical point to reproduce DRMM regards the set of word embeddings used as input to the model. Guo et al. share a word2vec model that has been trained on the collections used to evaluate DRMM [95]. However, to reproduce the experiments on Robust04 – and to generalize DRMM to other collections – we decided to train a new word2vec model following authors' steps and relying on Gensim. We train the model as described in Subsection 4.1.4. The word2vec hyperparameters are the same as those considered in the original experiments [95] but, since Guo et al. do not report the number of epochs used to train word2vec, we test different iterations and employ gradient decay from Gensim.

[5] The modified version of Terrier is available at the url: https://github.com/gridofpoints/

[6] http://www.lemurproject.org/galago.php

4.2.3 Experimental Results

In Table 4.3, we report the experimental results we obtained with the open-source DRMM code, provided by the authors, and our preprocessing pipeline. In particular, we compare the results obtained using the word embeddings from the word2vec model shared by Guo et al. (DRMM original) and those obtained using the word embeddings from the word2vec model we trained using Gensim (DRMM Gensim). We also compare the performances of our QLM implementation with Terrier 4.1 to those obtained by Guo et al [95].

Table 4.3 Results of the reproducibility study of DRMM. For each version of Robust04, the first row reports the scores of the original model version on MAP, nDCG@20, and P@20, the second row reports the scores of the reproduced version and the third row reports the difference between original and reproduced versions; a negative difference indicates that the reproduced versions are stronger than those originally used by the authors. **Bold** values represent the best model (original and reproduced), whereas *italic* values represent differences greater than 0.02 scores among the models. In both DRMM original and DRMM Gensim, the first row refers to the results presented by Guo et al. [95]

		Robust04 (t)			Robust04 (d)		
		MAP	nDCG@20	P@20	MAP	nDCG@20	P@20
QLM(dir)	orig.	0.253	0.415	0.369	0.246	0.391	0.334
	repr.	0.248	0.415	0.355	0.246	0.392	0.326
	diff.	+0.005	0.000	+0.014	0.000	-0.001	+0.008
DRMM original	orig.	**0.279**	**0.431**	**0.382**	**0.275**	**0.437**	**0.371**
	repr.	**0.270**	**0.442**	**0.377**	**0.252**	**0.415**	**0.347**
	diff.	+0.009	-0.011	+0.005	+*0.023*	+*0.022*	+*0.024*
DRMM Gensim	orig.	**0.279**	**0.431**	**0.382**	**0.275**	**0.437**	**0.371**
	repr.	0.268	0.441	0.376	0.249	0.411	0.343
	diff.	+0.011	-0.010	+0.006	+*0.026*	+*0.026*	+*0.028*

The results in Table 4.3 indicate that our preprocessing pipeline, along with the training strategy we employed to train word2vec models, lead to performances similar to those presented by Guo et al. [95]. Overall, we obtain MAP, nDCG@20, and P@20 scores close to those obtained by Guo et al. – with larger differences when considering the *description* (d) field. It is worth mentioning that if we repeat the experiments considering *title+description* (t+d), we obtain higher scores than those reported in Table 4.3. In particular, we obtain MAP, nDCG@20, and P@20 scores of 0.279, 0.451, and 0.386, respectively. In light of these results, we consider our preprocessing pipeline and training strategy reliable enough to reproduce DRMM performances.

4.2.4 Discussion

The experimental results highlighted a sizable impact of the word embeddings on the performances of DRMM. We stress that a detailed description of the word2vec training process lacks in the original DRMM paper [95]. Indeed, we had to perform numerous experiments and try different parameter combinations to train a word2vec model that would lead DRMM to performances close to those obtained by Guo et al. First, we trained word2vec using the official Google package and the hyperparameters configuration described in Subsection 4.1.4 – i.e., word2vec (corpus).[7] With this setting, we obtained a MAP value of 0.249 on Robust04 (t). Then, we relied on Gensim to train word2vec and, with the same hyperparameters configuration, we obtained a MAP value of 0.268. Thus, we adopted Gensim for all the experiments. However, it becomes clear that sharing the training strategy and the library used for word embeddings is fundamental to reproduce the results of a neural IR model.

Another issue we encountered regards the preparation of the input data required by DRMM. The available implementation of DRMM requires seven files: a run in TREC format (to be re-ranked); a file with trained word embeddings; a file containing the document and corpus frequency for each term in the collection; a file containing each document of the corpus with its identifier (the same used in the input run), its length, and the frequency of each term in it; a file with the ideal discounted cumulative gain value for each considered topic; a file with the list of terms for each topic, along with the topic identifier (the same used in the input run); a file with the relevance judgments in TREC format for the given topics and documents in the collection. However, the authors do not share a tool, or describe with enough detail the process employed to prepare the input data in such format. Therefore, we had to make a few assumptions about the preprocessing steps to be applied to Robust04. For instance, applying whitespace tokenization, stopwords removal, and stemming on the raw documents – as indicated in the original paper [95] – increased the noise introduced in the model. Thus, we first removed tags and punctuation, and then we performed tokenization, stopwords removal, and stemming.

For these reasons, we believe that sharing also the code and the libraries used to preprocess collections and train word embeddings would be good practice. Preprocessing is often underspecified – despite its sizable impact on the overall performances of IR models, and in particular of neural IR models.

[7] https://code.google.com/archive/p/word2vec/

4.3 Reproducibility Study: NVSM

We describe and reproduce the Neural Vector Space Model (NVSM), an end-to-end unsupervised representation-based neural model for ad hoc retrieval proposed by Van Gysel et al. [223]. As shown in Subsection 3.4.2, representation-based models learn (or exploit) distributed representations which, depending on the task, are used to perform semantic similarity between pairs of words or semantic matching between queries and documents. Within this framework, NVSM learns word and document representations to perform semantic matching between queries and documents. NVSM is one of the very few existing end-to-end unsupervised neural IR models and has great potential to generalize, as it does not require any interaction or labeled data for training.

4.3.1 The Neural Vector Space Model

NVSM jointly learns distinct word and document representations by optimizing an unsupervised loss function which minimizes the distance between sequences of n words (i.e. n-grams) and the documents containing them. Such optimization objective imposes that n-grams extracted from a document should be predictive of that document. Unlike the models from which it derives [222, 224], NVSM integrates a notion of term specificity [211, 190] in the learning process of word and document representations. In fact, while optimizing the n-gram representations to be close to the corresponding documents, words that are discriminative for the target documents learn to contribute more to the n-gram representations. Therefore, words associated with many documents will be neglected due to low predictive power. After training, the learned word and document representations are used to perform retrieval. Queries are seen as n-grams and matched against documents in the feature space. Documents are then ranked in decreasing order of the cosine similarity computed between query and document representations. Note that NVSM performs retrieval and then ranking on the whole document collection. Below, we provide a detailed description of NVSM components.

Given a document collection D and a word vocabulary V, the model considers the vector representations $\{\boldsymbol{w}_i\}_{i=1}^{|V|} \in \mathbb{R}^{|V|\times a}$ and $\{\boldsymbol{d}_j\}_{j=1}^{|D|} \in \mathbb{R}^{|D|\times b}$ for vocabulary words V and documents D, respectively, where a and b denote the dimensionality of word and document representations. Due to the different dimensionality of word representations $\boldsymbol{w}_i$ and document representations $\boldsymbol{d}_j$, the model requires a transformation $f : \mathbb{R}^a \to \mathbb{R}^b$ from the word feature space to the document feature space. The considered transformation is linear:

$$f(\boldsymbol{x}) = \boldsymbol{W}\boldsymbol{x} \tag{4.3}$$

where $\boldsymbol{x}$ is a a-dimensional vector and $\boldsymbol{W}$ is a $b \times a$ parameter matrix that is learned using gradient descent. A sequence of n words extracted from d and starting at position h (i.e., an n-gram) is defined as $S_h^n(d) = (w_i)_{i=h}^{h+n-1}$. Then, the representation of such sequence $S_h^n(d)$ is obtained by averaging its constituent word representations as follows:

$$g\left(S_h^n(d)\right) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{w}_i \tag{4.4}$$

Representations of words and documents are learned using mini-batches B of n-gram/-document pairs such that an n-gram representation is projected close to the document containing it. During training, an auxiliary function that L2-normalizes a vector of arbitrary dimensionality is further introduced:

$$\mathrm{norm}(\boldsymbol{x}) = \frac{\boldsymbol{x}}{||\boldsymbol{x}||} \tag{4.5}$$

Therefore, the projection of an n-gram into the b-dimensional document feature space can be written as the following composite function:

$$\boldsymbol{h}_h^n(d) = (f \circ \mathrm{norm} \circ g)(S_h^n(d)) \tag{4.6}$$

By estimating the per-feature sample mean and variance over batch B, the standardized projection of the n-gram representation is obtained as follows:

$$\overline{\boldsymbol{h}}_h^n(d) = \text{hard-tanh}\left(\frac{\boldsymbol{h}_h^n(d) - \hat{\mathbb{E}}[\boldsymbol{h}_h^n(d)]}{\sqrt{\hat{\mathbb{V}}[\boldsymbol{h}_h^n(d)]}} + \boldsymbol{\beta}\right) \tag{4.7}$$

The n-gram representation is optimized to be close to the corresponding document. The composition function $g(\cdot)$, in combination with the L2-normalization $\mathrm{norm}(\cdot)$, causes words to compete for contributing to the resulting n-gram representation. Therefore, words that are discriminative for the target document learn to contribute more to the n-gram representation, and consequently, the L2-norm of the representations of discriminative words is larger than the L2-norm of non-discriminative words. This incorporates a notion of term specificity into the model. Moreover, standardization forces n-gram representations to distinguish themselves solely in the dimensions that matter for matching.

The similarity between a document d and a sequence $S_h^n(d)$ in the latent vector space is defined as:

$$P(y|d, S_h^n(d)) = \sigma(\boldsymbol{d} \cdot \overline{\boldsymbol{h}}_h^n(d)) \tag{4.8}$$

where $\overline{\boldsymbol{h}}_h^n(d)$ is the standardized n-gram representation, $\sigma(\cdot)$ denotes the sigmoid function, and y is a binary indicator that states whether the representation of document d is similar to the projection of its n-gram $S_h^n(d)$ or not. The probability of a document d, given its n-gram $S_h^n(d)$, is then approximated by uniformly sampling t contrastive examples [98]:

$$\log \overline{P}(d|S_h^n(d)) = \frac{t+1}{2t}\Big(t \log P(y|d, S_h^n(d))) + \sum_{\substack{k=1,\\ d_k \sim \mathcal{U}(D)}}^{t} \log(1.0 - P(y|d_k, S_h^n(d)))\Big) \quad (4.9)$$

where $\mathcal{U}(D)$ represents the uniform distribution over documents D used to obtain the t contrastive examples. Then, the loss function used to optimize the model, averaged over the instances in the batch B, is:

$$L(\Theta|B) = -\frac{1}{|B|}\sum_{j=1}^{|B|} \log \overline{P}(d_j|S_h^n(d_j)) + \frac{\gamma}{2|B|}\Big(\sum_{i=1}^{|V|} ||\boldsymbol{w}_i||_2^2 + \sum_{j=1}^{|D|} ||\boldsymbol{d}_j||_2^2 + ||\boldsymbol{W}||_F^2\Big) \quad (4.10)$$

where Θ is the set of parameters $\{\{\boldsymbol{w}_i\}_{i=1}^{|V|}, \{\boldsymbol{d}_j\}_{j=1}^{|D|}, \boldsymbol{W}, \boldsymbol{\beta}\}$ and γ is a weight regularization hyperparameter.

After training, a query q is projected into the document feature space by the composition of f and g: $\boldsymbol{q} = \boldsymbol{W} \cdot \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{w}_i$. The matching score between a document d and a query q is given by the cosine similarity between their representations in the document feature space. Thus, documents are ranked in decreasing order according to their matching score with the query q.

4.3.2 Implementation Details

Retrieval Models

NVSM. We re-implement NVSM in Python, relying on TensorFlow to build its neural architecture. We use Whoosh to index the considered document collections.[8] Whoosh is a fast search engine library in Python that provides easy access to the underlying tokenized documents – similarly to `pyndri` [225], used in the original paper. We decided to rely on Whoosh instead of `pyndri` to keep the entire pipeline in Python. Otherwise, to use `pyndri`, we should first have had indexed the collections using Indri [215] – which is built in C++.[9] As in the original paper, we remove stopwords using the Indri stoplist,[10] and we do not perform stemming.

[8] https://whoosh.readthedocs.io/en/latest/

[9] https://www.lemurproject.org/indri/

[10] http://www.lemurproject.org/stopwords/stoplist.dft

One of the biggest challenges we found to reproduce the results obtained by Van Gysel et al. [223] regards the (hyper) parameter tuning. The original paper and the associated GitHub repository lack a comprehensive description for all the (hyper) parameters of NVSM – which are fundamental to reproduce its performances.[11] Therefore, to select the parameters and hyperparameters required for this study, we relied on the original paper [223], the authors' GitHub repository, and a previous seed work that NVSM extends [222]. For each (hyper) parameter, the reference sources are reported below.

words that have a document frequency greater than 1 and lower than or equal to $\frac{|\Phi|}{2}$

Word vocabulary:

- Vocabulary size is limited to 2^{16} words (GitHub repository: /scripts/functions.sh, seed paper [222]), or to $60,000$ words (original paper [223], GitHub repository: /cpp/main.cu).
- Words containing numbers are not considered (GitHub repository: /cpp/main.cu, seed paper [222]).
- Words with a document frequency lower than 2 and greater than $\frac{|D|}{2}$ are not considered (GitHub repository: /cpp/main.cu).

Model parameters:

- Pseudo-random number generator seed equal to 0 (GitHub repository: /cpp/main.cu).
- The number of batches for a single epoch is computed as $\lceil \frac{1}{|B|} \sum_{d \in D} (|d| - n + 1) \rceil$ (original paper [223]).
- Word representations, document representations and the parameter matrix W are uniformly sampled in the range $[-\sqrt{\frac{6.0}{m+n}}, \sqrt{\frac{6.0}{m+n}}]$ for an $m \times n$ matrix – following the initialization scheme presented by Glorot and Bengio [88] (GitHub repository, seed paper [222]).

Model hyperparameters: (all in the original paper [223])

- Word representation size $a = 300$.
- Document representation size $b \in \{64, 128, 256\}$.
- n-gram size $n \in \{4, 6, 8, 10, 12, 16, 24, 32\}$.
- Batch size equal to 51200.
- Number of training epochs equal to 15.
- Number of negative examples $t = 10$.
- Adam optimizer with parameters $\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$ (original paper [223] reports only α, seed paper [222] reports also β_1 and β_2).

[11] https://github.com/cvangysel/cuNVSM/

- Regularization $\gamma = 0.01$.

For each collection, the given set of topics is split into validation and test sets.[12] The sizes of document representations $b \in \{64, 128, 256\}$ and n-grams $n \in \{4, 6, 8, 10, 12, 16, 24, 32\}$ are optimized on the validation set and then used on the test set. The optimal hyperparameter combinations are not reported by Van Gysel et al.

In our experiments, we set the vocabulary size to 2^{16} and we select $b = 256$ – according to the results reported in Figure 3 of the original paper [223]. Similarly, we choose the n-gram size that provides the best scores in terms of MAP. We keep the rest of the parameters and hyperparameters as described above. We train NVSM for 15 epochs and we select the model iteration that performs best in terms of MAP on the validation set. The best model iteration is then evaluated on the test set. Table 4.4 shows the comparison between the results obtained with the original and reproduced versions of NVSM. Furthermore, for each collection considered, we indicate the optimal n-gram size and epoch at which we obtain the best NVSM iteration in Table 4.5.

Baselines. We reproduce word2vec (add), word2vec (si), and LDA as semantic baselines. Following Van Gysel et al. [223], we rely on Gensim to implement word2vec (skip-gram architecture) and LDA. For word2vec approaches, we adopt the same choices made for NVSM regarding word vocabulary, seed value, negative examples, one-sided window size (i.e., n-gram size $n/2$), and number of epochs. The embedding size is set to 256 to be consistent with NVSM. Documents are ranked in decreasing order of the cosine similarity between their representations and the query representation – where the query representation is the average of its word representations. Once again, we select the model iteration that performs best in terms of MAP on the validation set and we evaluate it on the test set. For LDA, we set the number of topics $K = 256$ and $\alpha = \beta = 0.1$. LDA is trained until topic convergence is achieved. At query time, documents are ranked in decreasing order of the cosine similarity between query and document topic distributions.

Regarding the lexical baselines, we reproduce QLM (jm) and QLM (dir). We rely on Indri [215] to index and query the considered collections (as in the original paper). For consistency with the semantic models, stopwords are removed using the Indri stoplist and no stemming is performed. The smoothing hyperparameters $\lambda \in \{x | k \in \mathbb{N}^+, k \leq 20, x = k/20\}$ and $\mu \in \{125, 250, 500, 750, 1000, 2000, 3000, 4000, 5000\}$ of QLM (jm) and QLM (dir), respectively, are optimized on the validation set. The comparison between the results obtained with the original and reproduced baselines is also shown in Table 4.4.

[12] Splits can be found at: https://github.com/cvangysel/cuNVSM/tree/master/resources/adhoc-splits/

Rank Fusion Models

We reproduce the fusion of three individual rankers: QLM (dir), word2vec (si), and NVSM. The fusion of these models provides a mixture of lexical and semantic matching signals. Van Gysel et al. [223] combined the individual rankers linearly and optimized the feature weights through a grid search using 20-fold cross-validation on the topic test sets. Feature weights are swept between 0 and 1 with increments of 0.0125 on each training fold. Individual features are normalized per query so that their values lie between 0 and 1. The configuration of the coefficients that achieves the highest MAP on the training set is selected and used to score the test fold. When scoring the test fold, the pool obtained from the top 1000 documents retrieved by the individual rankers is used as the candidate set.

However, due to the extensive memory/time requirements demanded by such an approach, the machine we used to run the experiments – a 2018 Alienware Area-51 with 36 cores and 64Gb of RAM – could not even finish the first training fold when we considered the entire document collection and an incremental step of 0.0125. Thus, we limited the number of documents in each fold of the training set to the pool of the top 1000 documents retrieved by the individual rankers. Also, we swept the feature weights between 0 and 1 with increments of 0.1, which is the minimum step that the available machine can handle.

Along with the supervised approach performed by Van Gysel et al. [223], we also employ three classic, efficient rank fusion methods [200]: CombSUM, CombMNZ, and CombANZ. In Table 4.6, we evaluate the results obtained with the supervised approach and the three classic methods on Robust04. The approach that performs best in MAP is then applied to all the considered collections and compared to the supervised approach from the original paper, as shown in Table 4.7.

4.3.3 Experimental Results

We present the results of the comparison between the original and the reproduced versions of QLM, LDA, word2vec, and NVSM in Table 4.4. For each collection, we report the results originally obtained by Van Gysel et al. [223], the results we obtain with the reproduced models, and the difference between original and reproduced versions.

Table 4.4 shows that the results obtained with the reproduced version of NVSM are close to those reported in the original paper. When we consider the performance difference, the only measure that presents an absolute difference greater than 0.02 is P@10 in WSJ (-0.038). In AP88-89 and NY collections, the absolute differences are lower than 0.01 for all measures. In terms of effectiveness, NVSM outperforms the semantic baselines in all the

Table 4.4 Retrieval results and comparison between original and reproduced versions of QLM, LDA, word2vec, and NVSM. For each considered collection, the first row reports the scores of the original model version on MAP, nDCG@100, and P@10, the second row reports the scores of the reproduced version, and the third row reports the difference between original and reproduced versions; a negative difference indicates that the reproduced versions are stronger than those originally used by the authors. **Bold** values represent the best model (original and reproduced), whereas *italic* values represent differences greater than 0.02. A two-tailed paired Student's t-test is computed between word2vec (si) and NVSM. Statistical significance is marked as * for $p < 0.1$, ** for $p < 0.05$ and *** for $p < 0.01$.

		AP88-89 (t)			FT (t)			LA (t)		
		MAP	nDCG@100	P@10	MAP	nDCG@100	P@10	MAP	nDCG@100	P@10
QLM (jm)	orig.	0.199	0.346	0.365	0.218	0.356	0.283	0.182	0.331	0.221
	repr.	0.199	0.346	0.364	0.209	0.337	0.258	0.178	0.319	0.214
	diff.	0.000	0.000	+0.001	+0.009	+0.019	*+0.025*	+0.004	+0.012	+0.007
QLM (dir)	orig.	0.216	0.370	0.392	**0.240**	**0.381**	**0.296**	**0.198**	**0.348**	**0.239**
	repr.	0.217	0.368	0.397	**0.230**	**0.362**	**0.270**	**0.198**	**0.341**	**0.233**
	diff.	-0.001	+0.002	-0.005	+0.01	+0.019	*+0.026*	0.000	+0.007	+0.006
LDA	orig.	0.039	0.077	0.078	0.009	0.028	0.013	0.004	0.015	0.010
	repr.	0.052	0.091	0.077	0.013	0.026	0.015	0.007	0.028	0.015
	diff.	-0.013	-0.014	+0.001	-0.004	+0.002	-0.002	-0.003	-0.013	-0.005
word2vec (add)	orig.	0.216	0.370	0.393	0.125	0.230	0.195	0.105	0.212	0.159
	repr.	0.234	0.395	0.416	0.140	0.252	0.214	0.075	0.165	0.116
	diff.	-0.018	*-0.025*	*-0.023*	-0.015	*-0.022*	-0.019	*+0.030*	*+0.047*	*+0.043*
word2vec (si)	orig.	0.230	0.383	0.418	0.141	0.250	0.204	0.131	0.242	0.179
	repr.	0.240	0.400	0.419	0.148	0.261	0.226	0.109	0.215	0.172
	diff.	-0.010	-0.017	-0.001	-0.007	-0.011	*-0.022*	*+0.022*	*+0.027*	+0.007
NVSM	orig.	**0.257****	**0.418****	**0.425**	0.172**	0.302***	0.239*	0.166**	0.300***	0.209*
	repr.	**0.257**	**0.414**	**0.429**	0.175**	0.304***	0.220	0.180***	0.316***	0.208**
	diff.	0.000	+0.004	-0.004	-0.002	-0.002	+0.019	-0.014	-0.016	+0.001

		NY (t)			Robust04 (t)			WSJ (t)		
		MAP	nDCG@100	P@10	MAP	nDCG@100	P@10	MAP	nDCG@100	P@10
QLM (jm)	orig.	0.158	0.270	0.376	0.201	0.359	0.369	0.175	0.315	0.345
	repr.	0.180	0.292	0.382	0.199	0.351	0.358	0.178	0.319	0.347
	diff.	*-0.022*	*-0.22*	-0.006	+0.002	+0.008	+0.011	-0.003	-0.004	-0.002
QLM (dir)	orig.	**0.188**	**0.318**	**0.486**	**0.224**	**0.388**	**0.415**	0.204	**0.351**	**0.398**
	repr.	**0.213**	**0.343**	**0.500**	**0.222**	**0.376**	**0.411**	0.205	0.355	0.391
	diff.	*-0.025*	*-0.025*	-0.014	+0.002	+0.012	+0.004	-0.001	-0.004	+0.007
LDA	orig.	0.009	0.027	0.022	0.003	0.010	0.009	0.038	0.082	0.076
	repr.	0.024	0.053	0.064	0.004	0.015	0.014	0.041	0.074	0.060
	diff.	-0.015	*-0.026*	*-0.042*	-0.001	-0.005	-0.005	-0.003	+0.008	+0.016
word2vec (add)	orig.	0.081	0.160	0.216	0.075	0.177	0.194	0.175	0.322	0.372
	repr.	0.100	0.196	0.252	0.065	0.158	0.184	0.181	0.326	0.393
	diff.	-0.019	*-0.036*	*-0.036*	+0.010	+0.019	+0.010	-0.006	-0.004	*-0.021*
word2vec (si)	orig.	0.092	0.173	0.220	0.093	0.208	0.234	0.185	0.330	0.391
	repr.	0.113	0.209	0.284	0.083	0.192	0.214	0.190	0.336	0.393
	diff.	*-0.021*	*-0.036*	*-0.064*	+0.010	+0.016	*+0.020*	-0.005	-0.006	-0.002
NVSM	orig.	0.117	0.208	0.296*	0.150***	0.287***	0.298***	**0.208****	**0.351**	0.370
	repr.	0.110	0.205	0.290	0.138***	0.270***	0.289***	**0.213*****	**0.359*****	**0.408**
	diff.	+0.007	+0.003	+0.006	+0.012	+0.017	+0.009	-0.005	-0.008	*-0.038*

considered collections but NY, where word2vec (si) shows better performances for MAP and nDCG@100. In particular, NVSM achieves the best results in AP88-89 and WSJ collections.

Regarding semantic baselines, the reproduced LDA performs similarly to the original one in all the considered collections. The most notable exceptions are nDCG@100 and P@10 in the NY collection, where reproduced LDA outperforms the original one with absolute differences of 0.026 and 0.042, respectively. For word2vec approaches, the reproduced word2vec (add) outperforms the original version in AP88-89, FT, WSJ, and NY for all the evaluation measures considered, whereas original word2vec (add) achieves better results in LA and Robust04. As for the performance difference, there is a marked gap between the two versions only in LA and NY collections – with differences of $+0.030$ for MAP, $+0.047$ for nDCG@100, and $+0.043$ for P@10 in LA, and of -0.036 for nDCG@100 and -0.036 for P@10 in NY. A similar trend can also be observed for word2vec (si), where the reproduced version outperforms the original one in AP88-89, FT, WSJ, and NY, whereas the original version achieves better results in LA and Robust04. The absolute differences between the two versions are lower than or close to 0.02 in all collections, except for NY and LA. In NY, the absolute differences are higher than 0.02 for all the considered measures, with a difference of -0.064 in P@10 – which is the highest (absolute) difference among all measures and collections. It is also worth mentioning that the reproduced word2vec (si) achieves a score of 0.284 for P@10 in NY, thus closing the gap with NVSM and resulting in a competitive semantic baseline.

As for lexical baselines, we observe that original and reproduced versions of QLM (jm) and QLM (dir) perform similarly in AP88-89, LA, WSJ, and Robust04 – where an absolute difference greater than 0.01 can be found between QLM (dir) versions only for nDCG@100 in Robust04, and between QLM (jm) versions only for nDCG@100 and P@10 in LA and Robust04, respectively. Larger differences between QLM (jm) and QLM (dir) versions can be observed in FT and NY collections. In FT, there is a marked difference between QLM original and reproduced versions for nDCG100 and P@10. Original QLM (jm) outperforms the reproduced one by an absolute difference of 0.019 for nDCG@100 and of 0.025 for P@10. Similarly, original QLM (dir) outperforms the reproduced one by an absolute difference of 0.019 for nDCG@100 and of 0.026 for P@10. On the other hand, we observe an opposite behavior in NY, where reproduced versions of QLM outperform original ones in all measures – with absolute differences greater than 0.02 for MAP and nDCG@100. In terms of effectiveness, QLM (dir) achieves the best results for all the considered measures in FT, LA, NY, and Robust04. In WSJ, instead, original QLM (dir) achieves the best results for nDCG@100 and P@10, but its reproduced version is outperformed by NVSM.

Thus, the results reported in Table 4.4 show that we successfully reproduced NVSM and that the original and reproduced versions of the baselines are aligned.

Table 4.5 NVSM optimal n-gram size and best epoch for each collection.

	n-gram size	Best epoch
AP88-89	16	13
FT	16	11
LA	10	10
WSJ	16	14
Robust04	16	11
NY	16	1

Table 4.6 shows the results of the supervised and classic rank fusion approaches in Robust04. For each combination of QLM (dir) with word2vec (si) and NVSM, we observe that the supervised approach performs consistently worse than its original version. Since the original and reproduced versions of QLM (dir) perform similarly, this indicates that we did not reproduce the supervised rank fusion method successfully. Overall, the best rank fusion method is CombSUM, while the worst is CombANZ. Therefore, we use CombSUM in all the considered collections and we compare it to the original version of the supervised rank fusion method in Table 4.7.

In Table 4.7, we observe that AP88-89 is the only collection where CombSUM achieves results close to those of the supervised method – with differences between the two versions of QLM(dir)+word2vec(si)+NVSM of $+0.015$ for MAP, $+0.005$ for nDCG@100, and $+0.005$ for P@10. Overall, CombSUM shows performance gains in all the combinations of the individual rankers in AP88-89, FT, and WSJ. The performance gains on LA are positive for QLM(dir)+NVSM and QLM(dir)+word2vec(si)+NVSM, whereas they are negative for QLM(dir)+word2vec(si). In particular, the CombSUM of QLM(dir)+NVSM consistently outperforms the original supervised method in all the measures considered. In Robust04, the only combination that improves over the baseline is QLM(dir)+NVSM. However, when compared to the original supervised method, the CombSUM of QLM(dir)+NVSM presents a performance gap of about 0.020 in all the measures. Regarding NY, each combination of individual rankers using CombSUM achieves lower performances than the baseline. In particular, the difference between the original supervised method and the CombSUM of QLM(dir)+word2vec(si)+NVSM is $+0.100$ for P@10.

4.3.4 Discussion

The results presented in Table 4.4 show that the original and reproduced versions of NVSM perform similarly. This indicates that we successfully reproduced the results obtained by

Table 4.6 Results and comparison between the supervised rank fusion method and CombSUM, CombMNZ and CombANZ in Robust04. The first two rows report, for reference, the scores of the original and reproduced versions of QLM (dir) for MAP, nDCG@100, and P@10. For each combination of QLM (dir) with word2vec (si) and NVSM, the first two rows report the scores of the original and reproduced versions of the supervised method. Then, subsequent rows report the scores of CombSUM, CombMNZ, and CombANZ, respectively. **Bold** values represent the best method among the supervised and the three classic methods.

		Robust04 (t)		
		MAP	nDCG@100	P@10
QLM (dir)	orig.	0.224	0.388	0.415
	rep.	0.222	0.376	0.411
QLM+word2vec	orig.	0.232	0.399	0.428
	repr.	0.190	0.344	0.346
	CSUM	**0.199**	**0.358**	**0.378**
	CMNZ	0.191	0.353	0.366
	CANZ	0.179	0.325	0328
QLM+NVSM	orig.	0.247	0.411	0.448
	repr.	0.204	0.357	0.375
	CSUM	**0.230**	0.391	**0.424**
	CMNZ	0.229	**0.392**	0.419
	CANZ	0.199	0.352	0.373
QLM+word2vec+NVSM	orig.	0.247	0.412	0.446
	repr.	0.1836	0.336	0.344
	CSUM	**0.206**	**0.368**	**0.385**
	CMNZ	0.202	0.363	0.373
	CANZ	0.175	0.323	0.338

Van Gysel et al. with NVSM [223]. However, we had to look into different sources to get the appropriate (hyper) parameters – that is, the original paper [223], the GitHub repository, and a previous seed paper [222]. Besides, the lack of information regarding the optimal hyperparameters used to obtain the results presented in Table 2 of the original paper led us to identify them differently. In fact, we relied on Figure 3 of the original paper to obtain the optimal hyperparameters.

Another critical aspect we encountered while reproducing NVSM regards the lexicon. For example, NVSM does not retrieve any document from Robust04 for the following four topics: topic 312 "*Hydroponics*", topic 316 "*Polygamy Polyandry Polygyny*", topic 348 "*Agoraphobia*", and topic 379 "*mainstreaming*". If we analyze the content of such topics, we

Table 4.7 Rank fusion results and comparison between CombSUM and the supervised rank fusion method adopted by Van Gysel et al. [223]. For each considered collection, the scores of the original and reproduced versions of QLM (dir) for MAP, nDCG@100, and P@10 are reported for reference. For each combination of QLM (dir) with word2vec (si) and NVSM, the first row reports the scores of the original supervised rank fusion, the second row reports the scores of the CombSUM rank fusion, and the third row reports the difference between the original supervised approach and CombSUM; a negative difference indicates that CombSUM achieves higher scores than those of the original supervised approach. **Bold** values represent the best method (original and reproduced), whereas *italic* values represent differences greater than 0.02. The percentage gain (or loss) over the baseline method is reported next to each rank fusion approach.

		AP88-89 (t)			FT (t)		
		MAP	nDCG@100	P@10	MAP	nDCG@100	P@10
QLM (dir)	orig.	0.216	0.370	0.392	0.240	0.381	0.296
	repr.	0.217	0.368	0.397	0.230	0.362	0.270
QLM+word2vec	orig.	0.279 (+29%)	0.437 (+18%)	0.450 (+14%)	0.251 (+4%)	0.393 (+3%)	0.313 (+6%)
	CSUM	0.275 (+27%)	0.441 (+20%)	0.446 (+12%)	0.242 (+5%)	0.381 (+5%)	0.293 (+9%)
	diff.	+ 0.004	- 0.004	+ 0.004	+ 0.009	+ 0.012	+ *0.020*
QLM+NVSM	orig.	0.289 (+33%)	0.444 (+20%)	0.473 (+20%)	0.251 (+4%)	0.401 (+5%)	0.322 (+9%)
	CSUM	0.269 (+24%)	0.428 (+16%)	0.456 (+15%)	0.233 (+1%)	0.378 (+4%)	0.286 (+6%)
	diff.	+ *0.020*	+ 0.016	+ 0.017	+ 0.018	+ *0.023*	+ *0.036*
QLM+word2vec+NVSM	orig.	**0.307** (+42%)	**0.466** (+26%)	**0.498** (+27%)	**0.258** (+7%)	**0.406** (+6%)	**0.322** (+9%)
	CSUM	**0.292** (+35%)	**0.461** (+25%)	**0.493** (+24%)	**0.244** (+6%)	**0.386** (+7%)	**0.297** (+10%)
	diff.	+ 0.015	+ 0.005	+ 0.005	+ 0.014	+ *0.020*	+ *0.025*
		LA (t)			NY (t)		
		MAP	nDCG@100	P@10	MAP	nDCG@100	P@10
QLM (dir)	orig.	0.198	0.348	0.239	0.188	0.318	0.486
	repr.	0.198	0.341	0.233	**0.213**	**0.343**	**0.500**
QLM+word2vec	orig.	0.212 (+7%)	0.360 (+3%)	0.236 (-1%)	0.206 (+9%)	0.333 (+4%)	0.494 (+1%)
	CSUM	0.191 (-4%)	0.326 (-4%)	0.229 (-2%)	0.194 (-9%)	0.325 (-5%)	0.436 (-13%)
	diff.	+ *0.021*	+ *0.034*	+ 0.007	+ 0.012	+ 0.008	+ *0.058*
QLM+NVSM	orig.	0.220 (+11%)	0.376 (+7%)	0.244 (+1%)	**0.222** (+18%)	**0.355** (+11%)	0.520 (+6%)
	CSUM	**0.232** (+17%)	**0.381** (+12%)	**0.255** (+9%)	0.198 (-7%)	0.333 (-3%)	0.476 (-5%)
	diff.	- 0.012	- 0.005	- 0.011	+ *0.024*	+ *0.022*	+ *0.044*
QLM+word2vec+NVSM	orig.	**0.226** (+14%)	**0.378** (+8%)	**0.250** (+4%)	**0.222** (+18%)	0.353 (+10%)	**0.526** (+8%)
	CSUM	0.214 (+8%)	0.366 (+7%)	0.251 (+7%)	0.182 (-14%)	0.320 (-7%)	0.426 (-15%)
	diff.	+ 0.012	+ 0.012	- 0.001	+ *0.040*	+ *0.033*	+ *0.100*
		Robust04 (t)			WSJ (t)		
		MAP	nDCG@100	P@10	MAP	nDCG@100	P@10
QLM (dir)	orig.	0.224	0.388	0.415	0.204	0.351	0.398
	repr.	0.222	0.376	0.411	0.205	0.355	0.391
QLM+word2vec	orig.	0.232 (+3%)	0.399 (+2%)	0.428 (+2%)	0.254 (+24%)	0.410 (+16%)	0.454 (+13%)
	CSUM	0.199 (-10%)	0.358 (-5%)	0.378 (-8%)	0.238 (+16%)	0.399 (+12%)	0.449 (+15%)
	diff.	+ *0.033*	+ *0.041*	+ *0.050*	+ 0.016	+ 0.011	+ 0.005
QLM+NVSM	orig.	**0.247** (+10%)	0.411 (+6%)	**0.448** (+7%)	0.248 (+21%)	0.396 (+12%)	0.425 (+6%)
	CSUM	**0.230** (+4%)	**0.391** (+4%)	**0.424** (+3%)	0.244 (+19%)	0.403 (+14%)	0.443 (+13%)
	diff.	+ 0.017	+ *0.020*	+ *0.024*	+ 0.004	- 0.007	- 0.018
QLM+word2vec+NVSM	orig.	**0.247** (+10%)	**0.412** (+6%)	0.446 (+7%)	**0.271** (+32%)	**0.426** (+21%)	**0.456** (+14%)
	CSUM	0.206 (-7%)	0.369 (-2%)	0.385 (-6%)	**0.251** (+22%)	**0.416** (+17%)	**0.455** (+16%)
	diff.	+ *0.041*	+ *0.043*	+ *0.061*	+ *0.020*	+ 0.010	+ 0.001

see that three out of four are composed of a single word. Besides, the NVSM word vocabulary does not contain any of these terms. Therefore, NVSM cannot retrieve any document for them. Knowing the exact lexicon used in the original paper is pivotal to reproduce the results. Otherwise, we cannot understand whether the differences between original and reproduced versions are related to implementation nuances or different preprocessing steps.

Regarding semantic baselines, we followed the same setup presented by Van Gysel et al. for LDA [223]. However, the LDA implementation from Gensim presents many more parameters than those mentioned in the original paper. Thus, not knowing what values to assign to such parameters prevents the reproducibility of the results. On the other hand, we adopted the same hyperparameter choices of NVSM for word2vec approaches. The resulting versions of word2vec (add) and word2vec (si) present sizable differences with the original ones. Most likely, the choices we made are different than those made by Van Gysel et al. [223] – especially when we consider the results obtained with the reproduced word2vec approaches in NY. In the original paper there is no information about the optimal choices for word2vec approaches, nor any figure that can help to identify a subset of candidate choices. Moreover, the same considerations made about the lexicon for NVSM also hold for word2vec approaches. Nevertheless, the results of the two-tailed paired Student's t-tests between reproduced word2vec (si) and NVSM are consistent with those originally obtained by Van Gysel et al. The only notable exceptions can be found in AP88-89, where there is no statistical difference between the reproduced versions, and in WSJ, where there is a statistical difference between reproduced word2vec (si) and NVSM for nDCG@100.

As for lexical baselines, we employed the same search engine, that is Indri, and performed the same operations reported by Van Gysel et al. [223] for indexing, querying, and parameter tuning. Therefore, the differences between the original and reproduced versions of QLM (jm) and QLM (dir) might lie on the different tokenization process applied to topics. In fact, we relied on Indri for both indexing and querying, whereas Van Gysel et al. employ `pyndri` [225] for querying.

Concerning rank fusion, the main issues we found regard the extensive memory/time requirements demanded by the supervised rank fusion method. The choices we made to try reproducing this method were insufficient to obtain results comparable to those presented in the original paper. On the other hand, classic, efficient rank fusion methods like CombSUM produced mixed results – which even worsened the performances of QLM (dir) in some collections. Nevertheless, the trade-off between effectiveness and efficiency brought by CombSUM shows that far less expensive fusion methods can be used to improve performances, achieving – on some collections – performance gains similar to those obtained by Van Gysel et al. [223].

4.4 Comparison between Lexical and Semantic Models

We compare lexical and semantic models, with a focus on DRMM and NVSM. The objective of this evaluation is to investigate the potential and limitations of semantic models compared to lexical ones. By understanding the strengths and weaknesses of semantic models, we can improve their combination with lexical models and develop multi-stage IR systems that are more effective in addressing the semantic gap. The test collections and baselines considered are reported in Subsection 4.1.2. For semantic models, we use the model iterations that perform best in the experiments presented in Sections 4.2 and 4.3. In addition to comparing lexical and semantic models, we also investigate the effectiveness of unsupervised semantic models in re-ranking. To this end, we employ NVSM to re-rank the same QLM (dir) runs used by DRMM. Table 4.8 presents the results of this evaluation.

4.4.1 Experimental Results

The results in Table 4.8 show that lexical models outperform all semantic models in NY. Conversely, DRMM outperforms both lexical and semantic models in Robust04. On the other hand, NVSM performs better than word2vec approaches in both NY and Robust04, but it is never competitive with the lexical baselines nor DRMM. Regarding re-ranking, NVSM outperforms DRMM in NY but not in Robust04. Besides, we observe that using NVSM to re-rank QLM (dir) worsens QLM performances in both collections. This suggests that NVSM might grasp different signals than QLM, and thus not leveraging effectively the set of candidate documents retrieved by QLM – which relies on lexical matching signals.

For reference purposes, we report the best values for MAP and P@10 obtained in the TREC 2004 Robust Retrieval Track [232], and the best values for MAP in the TREC 2017 Common Core Track [12, 226]. In Robust04 (t), the best value for MAP is 0.333 and for P@10 is 0.513. In NY, the best value for MAP is 0.538.

4.4.2 Statistical Analysis

Figure 4.2 reports the results of the Tukey's HSD test on the runs produced by the models reported in Table 4.8. First of all, we observe that all the lexical models belong to the top group in NY. On the other hand, none of the semantic models – be it a retrieval or a re-ranking model – belong to the top group. Therefore, lexical models perform statistically better than semantic ones in NY. We have a different scenario in Robust04. In this case, DRMM belongs to the top group for all the considered measures and statistically outperforms the other semantic models – including QLM/NVSM, which performs the same task. However, it is

Table 4.8 Comparison between lexical and semantic models. The lexical models considered are: QLM (dir), BM25, TF-IDF, and DFR. The semantic retrieval models considered are: word2vec (add), word2vec (si), and NVSM. The semantic re-ranking models considered are: DRMM and QLM/NVSM. For each collection, the scores for MAP, nDCG@100, and P@10 are reported. **Bold** values represent the best scores among models.

	NY (t)			Robust04 (t)		
	MAP	nDCG@100	P@10	MAP	nDCG@100	P@10
QLM (dir)	0.232	0.370	**0.522**	0.248	0.411	0.424
BM25	**0.234**	0.347	0.468	0.242	0.404	0.431
TF-IDF	0.228	**0.499**	0.472	0.242	0.405	0.431
DFR	0.225	0.350	0.478	0.227	0.390	0.425
word2vec (add)	0.100	0.196	0.252	0.062	0.150	0.175
word2vec (si)	0.113	0.209	0.284	0.081	0.185	0.205
NVSM	0.110	0.205	0.290	0.139	0.269	0.279
DRMM	0.141	0.211	0.274	**0.268**	**0.435**	**0.455**
QLM/NVSM	0.152	0.252	0.328	0.157	0.289	0.287

worth mentioning that QLM/NVSM is an unsupervised re-ranking model, as opposed to DRMM. Therefore, QLM/NVSM does not exploit relevance judgments to learn how to rank documents given a query. Rather, QLM/NVSM simply leverages the word and document representations learned by NVSM to re-rank the initial set of candidate documents retrieved by QLM (dir).

4.4.3 Discussion

From the results reported in Table 4.8, we first observe that DRMM presents a large performance difference in the two considered collections. In Robust04, DRMM improves over QLM (dir) and achieves top results for all the evaluation measures. Conversely, DRMM worsens the ranking produced by QLM (dir) in NY. This result reflects one of the limitations of supervised neural IR models: the inability to generalize when trained on a limited number of topics. Indeed, compared to Robust04, which presents 249 topics, NY has only 50 topics – i.e., one fifth of the topics in Robust04. Therefore, DRMM suffers from the lack of topics to learn from in NY. In addition, the large number of documents contained within NY generates a wide variety of matching signals that DRMM needs to interpret. For this reason, DRMM struggles to learn how to discriminate between relevant and non-relevant documents when it observes only a small fraction of the collection – that is, the one related to the set of given topics. This intuition is also supported by the results obtained in the WT2g collection, as we

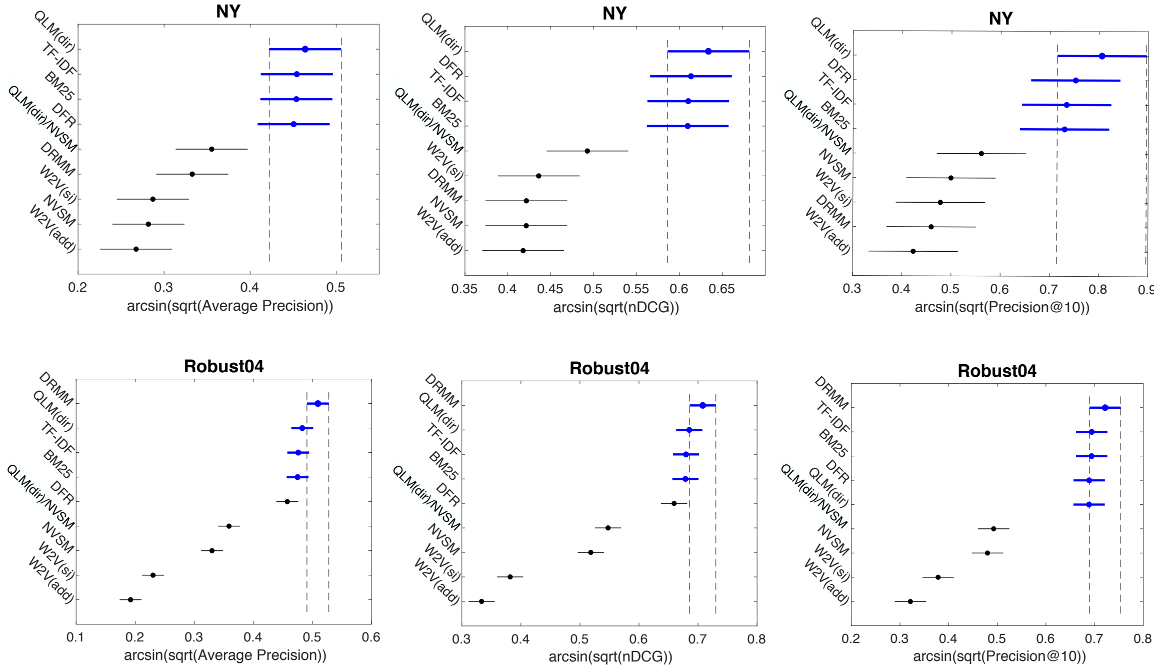


Fig. 4.2 Statistical tests for the results reported in Table 4.8 (top group highlighted). All pairwise comparisons are calculated with Tukey's HSD confidence intervals and a significance level $\alpha = 0.05$. Each row depicts the comparisons made for MAP, nDCG@100, and P@10 in a specific collection.

will see in Section 4.5. Indeed, WT2g has the same number of topics of NY, but it contains seven times fewer documents. The reduced corpus size compensates the lack of training topics for DRMM, which achieves similar results to lexical models in this collection.

Regarding unsupervised semantic models, we see that NVSM outperforms word2vec approaches in Robust04, whereas it achieves similar results to word2vec (si) in NY. In both collections, all the unsupervised semantic models do not compete with the lexical baselines and DRMM – although the difference with DRMM is small in NY. Furthermore, the re-ranking performed by NVSM worsens the performances of QLM (dir) in both collections. This suggests that NVSM grasps different signals than QLM. In other words, NVSM does not effectively leverage the set of documents retrieved by QLM since it performs semantic matching rather than lexical matching. Thus, rank fusion is better suited than re-ranking for NVSM, as lexical and semantic matching signals can be combined to retrieve relevant documents most affected by the semantic gap (as also shown in Table 4.7).

As for state-of-the-art approaches, BM25+RM3 [244] achieves a MAP value of 0.290 in Robust04,[13] while the BERT application developed by Yang et al. [246] for ad hoc retrieval

[13] https://github.com/castorini/anserini/blob/master/docs/experiments-robust04.md

achieves MAP of 0.328. Compared to these approaches, neither NVSM nor DRMM are competitive. Nevertheless, both NVSM and DRMM remain relevant models for a variety of different reasons. In particular, NVSM is still one of the most effective unsupervised neural models for ad hoc retrieval, whereas DRMM can integrate different – and more effective – word representations during training. Besides, the advances brought by DRMM and NVSM are methodological rather than performance-driven. Therefore, they can serve as core components for more advanced approaches, like the one presented by MacAvaney et al. [153] for supervised re-ranking or the one we present in Section 6.5 for unsupervised retrieval.

4.5 Collection-based Evaluation

We evaluate the impact that different domains and languages have on the performances of DRMM and NVSM. In other words, we evaluate the ability of DRMM and NVSM to generalize to other domains and tasks. The test collections and baselines considered are presented in Subsection 4.1.3.

4.5.1 Parameter Tuning

We optimize the number of bins in matching histograms and the size of the hidden layer of DRMM in OHSUMED and WT2g collections. OHSUMED is the test collection that differs the most from those used by Guo et al. [95], whereas WT2g represents a small-size Web collection similar to ClueWeb-09-Cat-B (which was used in the original DRMM paper).[14] For optimization, we consider the number of bins in matching histograms in the range $\{5,10,15,20,25,30,35\}$, the size of the hidden layer in the range $\{5,10,15,20,25\}$, and then we keep the combination that performs best. The results of this optimization process do not provide any (sizable) performance gain in OHSUMED, where the default values adopted by Guo et al. remain effective. On the other hand, a number of bins in matching histograms equal to 20 and a size for the hidden layer equal to 15 prove effective in WT2g – with performance gains of 0.02 or higher for MAP, nDCG100, and P10. Thus, we keep this combination for the rest of the experiments in WT2g.

Regarding NVSM, we optimize the following hyperparameters in each collection:

- Vocabulary size $|V| \in \{2^{16}, 2^{17}\}$;
- Document representation dimension $b \in \{128, 256\}$;
- n-gram size $n \in \{4,6,8,10,12,16,24,32\}$;

[14] https://lemurproject.org/clueweb09/

- Batch size in $\{12800, 25600, 51200\}$;
- Regularization $\gamma \in \{0.01, 0.1, 1.0\}$.

We keep the rest of the (hyper) parameters as in Section 4.3. Due to the prohibitive time required to perform grid search over the considered hyperparameters, we first optimize the n-gram size by keeping the default values for $|V| = 2^{16}$, $k_d = 256$, $|B| = 51200$, and $\gamma = 0.01$. Then, in each collection we select the n-gram size that performs best for MAP and we optimize the rest of the hyperparameters. From this optimization, none of the different combinations of document representation size, batch size, and regularization value proves better than the original setup. Conversely, the vocabulary size shows a significant impact on two collections: WT2g and CLEF-DE. In WT2g, NVSM improves from MAP: 0.206, nDCG@100: 0.356, and P@10: 0.370, with $|V| = 2^{16}$, to MAP: 0.225, nDCG@100: 0.380, and P@10: 0.402, with $|V| = 2^{17}$. Similarly, in CLEF-DE NVSM goes from MAP: 0.194, nDCG@100: 0.322, and P@10: 0.281, with $|V| = 2^{16}$, to MAP: 0.211, nDCG@100: 0.343, and P@10: 0.301, with $|V| = 2^{17}$. Thus, we adopt $|V| = 2^{17}$ in WT2g and CLEF-DE collections and we keep the rest of the (hyper) parameters as in Section 4.3. The optimal n-gram size, vocabulary size, and the epoch at which we obtain the best NVSM iteration are reported in Table 4.9 for every collection.

Table 4.9 NVSM optimal n-gram size, vocabulary size, and epoch for each collection.

	n-gram size	Vocabulary	Best epoch
WT2g	16	131,072	11
OHSUMED	16	65,536	12
CLEF-IT	20	65,536	14
CLEF-DE	8	131,072	13
CLEF-FA	16	65,536	15

4.5.2 Experimental Results

We present the results of the generalization of DRMM and NVSM to different domains and languages in Tables 4.10 and 4.11, respectively. In other words, Table 4.10 shows the results of the considered models for WT2g and OHSUMED collections, whereas Table 4.11 shows the results for CLEF collections.

Regarding domains, Table 4.10 highlights that DRMM always improves the ranking produced by QLM (dir) – which is used to retrieve the initial set of 2000 candidate documents for DRMM. As a side note, we also employed DRMM to re-rank the runs obtained with

Table 4.10 Results of the generalization experiments on different domains. The test collections considered are WT2g (Web) and OHSUMED (medicine). **Bold** values represent the best model. The evaluation measures considered are MAP, nDCG@100 and P@10.

	WT2g (t+d)			OHSUMED (d)		
	MAP	nDCG@100	P@10	MAP	nDCG@100	P@10
TF-IDF	0.234	0.389	0.419	0.250	0.370	0.364
QLM (dir)	0.264	0.418	0.418	0.212	0.326	0.305
BM25	**0.296**	0.454	**0.484**	0.250	0.369	0.364
DFR	0.267	0.426	0.452	0.236	0.354	0.344
DRMM	0.289	**0.455**	0.434	**0.272**	**0.407**	**0.375**
NVSM	0.225	0.380	0.402	0.214	0.335	0.319

TF-IDF and BM25, but we did not find any sizable improvement over such baselines. On the other hand, NVSM achieves the lowest scores in WT2g and outperforms only QLM (dir) in OHSUMED. However, NVSM does not rely on any interaction or labeled data and extends two models tailored to product and expert search [222, 224]. Therefore, NVSM has a robust domain-specific nature and, for heterogeneous collections like WT2g – where documents have different contents and scopes – it generalizes worse than for homogeneous collections like OHSUMED. Indeed, NVSM outperforms QLM (dir) and achieves closer results to BM25 and DFR for all the considered measures in OHSUMED.

For reference purposes, the best MAP score obtained in WT2g during the TREC-8 Web Track [104] is equal to 0.383.

Table 4.11 Results of the generalization experiments on different languages. The test collections considered are CLEF-IT (Italian), CLEF-DE (German), and CLEF-FA (Farsi). **Bold** values represent the best model. The evaluation measures considered are MAP, nDCG@100, and P@10.

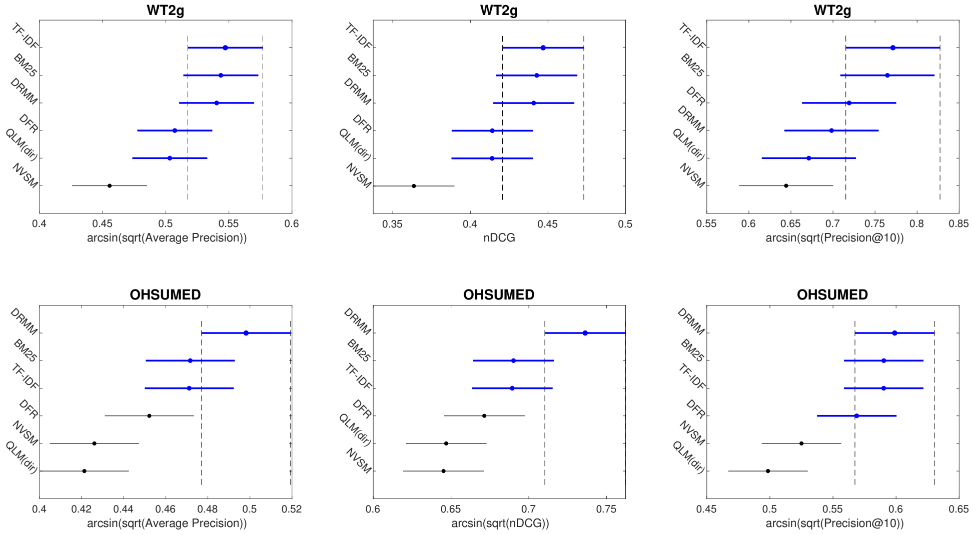
	CLEF-IT (t+d)			CLEF-DE (t+d)			CLEF-FA (t+d)		
	MAP	nDCG@100	P@10	MAP	nDCG@100	P@10	MAP	nDCG@100	P@10
TF-IDF	**0.438**	**0.628**	**0.408**	0.251	0.392	0.367	0.411	0.553	0.601
QLM (dir)	0.334	0.510	0.319	0.209	0.350	0.324	0.200	0.340	0.405
BM25	0.437	0.627	0.402	**0.253**	**0.395**	**0.373**	0.408	0.550	0.594
DFR	0.415	0.605	0.393	0.241	0.382	0.350	**0.421**	**0.565**	0.619
DRMM	0.357	0.557	0.349	0.188	0.329	0.316	0.375	0.551	**0.623**
NVSM	0.345	0.522	0.317	0.211	0.343	0.301	0.342	0.494	0.567

As for languages, Table 4.11 shows that DRMM achieves different results depending on the considered collection. Compared to QLM (dir), DRMM improves performances in CLEF-IT and CLEF-FA, but it worsens performances in CLEF-DE. Besides, the performance gains achieved by DRMM in CLEF-IT are small compared to those in CLEF-FA. On the other hand, NVSM shows competitive results with QLM (dir) in CLEF-DE and CLEF-IT, while it outperforms QLM (dir) in CLEF-FA. Compared to DRMM, NVSM shows close results in CLEF-IT, outperforms DRMM in CLEF-DE, but performs worse than DRMM in CLEF-FA. Overall, lexical baselines achieve the best results in all CLEF collections for all measures but P@10 in CLEF-FA – where DRMM performs best.

For reference purposes, we report below the best MAP values obtained in CLEF-DE and CLEF-IT during the CLEF 2006 Ad Hoc Track [60], and the best MAP value obtained in CLEF-FA during the CLEF 2009 Ad Hoc Track [81]. The best values for MAP in CLEF-DE, CLEF-IT, and CLEF-FA are, respectively, 0.419, 0.483, and 0.494.

4.5.3 Statistical Analysis

Figure 4.3 reports the results of the Tukey's HSD test on the runs produced to perform the generalization experiments presented in Tables 4.10 and 4.11. NVSM belongs to the top group only for P@10 in CLEF-FA. On the other hand, DRMM belongs to the top group for most of the considered measures and collections. The only notable exceptions are CLEF-DE and CLEF-IT, where DRMM does not belong to the top group for any of the considered measures. Regarding lexical baselines, TF-IDF, BM25, and DFR are generally the best performing models, whereas QLM (dir) is often the worst performing one. In particular, QLM (dir) does not belong to the top group for any of the considered measures in OHSUMED, CLEF-IT, and CLEF-FA.



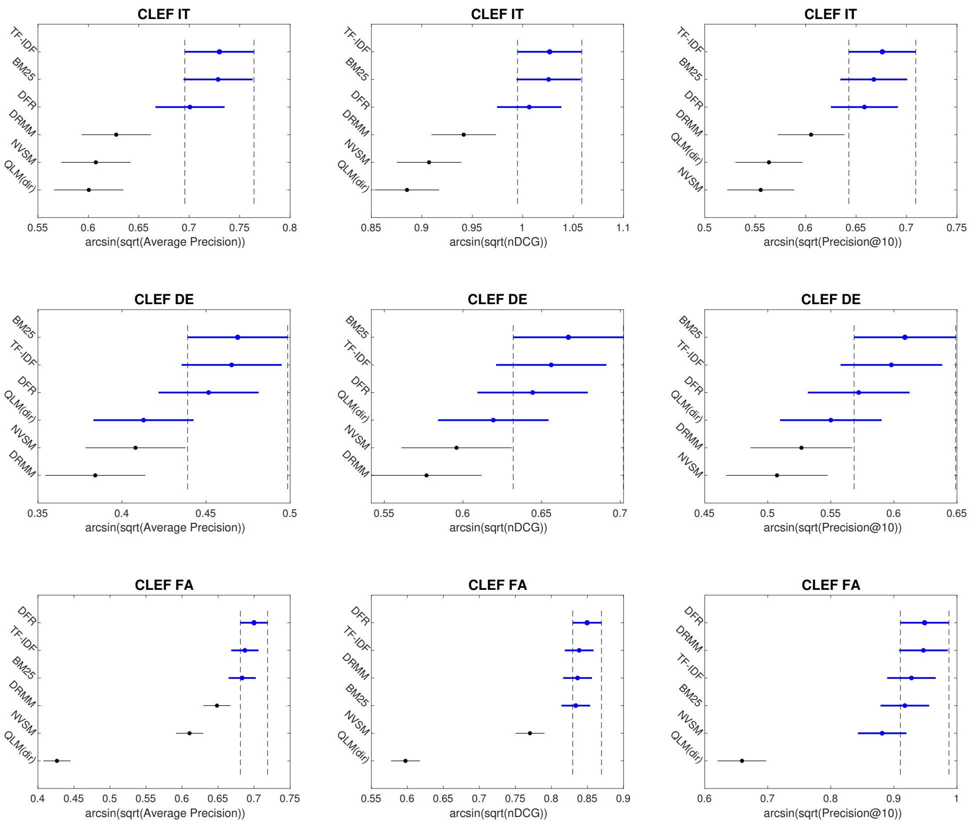


Fig. 4.3 Significance tests for the results of the generalization experiments reported in Tables 4.10 and 4.11 (top group highlighted). All pairwise comparisons are calculated with Tukey's HSD confidence intervals and a significance level $\alpha = 0.05$. Each row depicts the comparisons made for MAP, nDCG@100, and P@10 in a specific collection.

4.5.4 Discussion

The experimental results highlight the inconstancy of DRMM across the different test collections. In fact, the only collections where DRMM statistically improves the results obtained by QLM (dir) are OHSUMED and CLEF-FA. In other words, DRMM often fails to provide statistical improvements over the runs it re-ranks. Therefore, it is unclear whether the improvement brought by DRMM in re-ranking can be attributed to DRMM itself or to the lexical model used to produce the initial run – which might be good at finding relevant documents but bad at ranking them. In fact, the results indicate that DRMM achieves competitive performances when those of QLM (dir) are low, like in OHSUMED and CLEF-FA. Conversely, when QLM (dir) performs similarly to the other lexical models, like in WT2g and CLEF-DE, DRMM is less effective or even detrimental. In particular, DRMM worsens QLM (dir) results for all the considered measures in CLEF-DE. However, the overall results in CLEF-DE – compared to those obtained in the other CLEF collections – suggest that CLEF-DE is a difficult collection. Thus, we cannot exclude that with more topics for training, DRMM would improve the ranking produced by QLM (dir) also in this collection.

Regarding NVSM, the results indicate that it is not competitive with lexical models. The only exception is QLM (dir), that achieves performances comparable to or lower than those of NVSM in OHSUMED and CLEF collections. Also, the results in WT2g (Table 4.10) show that NVSM struggles to generalize to heterogeneous data. We believe that NVSM suffers more in heterogeneous collections like WT2g because of its inherent domain-specific nature. In fact, NVSM has been proposed for newswire retrieval and it extends two models tailored to product and expert search [222, 224], respectively.

4.6 Embedding-based Evaluation

We evaluate the effect that word embeddings – learned by different models – have on DRMM. DRMM can use different word embeddings, regardless of their characteristics. As opposed to NVSM, that jointly learns word and document representations, DRMM does not learn any representation. Rather, DRMM learns to interpret the interactions between documents and query terms, given the provided word embeddings. Therefore, we investigate how sensitive DRMM can be to different word embeddings.

We consider the word embeddings described in Subsection 4.1.4: (i) word2vec (corpus), word2vec (Google), fasttext, and NVSM. All the models but word2vec (Google) have been trained on the test collections where we perform this evaluation.

4.6.1 Experimental Results

The results from Table 4.12 highlight the robustness of DRMM to different word embeddings. There are no sizable differences between word2vec (corpus) and fasttext in Robust04, NY, and OHSUMED. However, fasttext embeddings lead to better performances in NY and WT2g. Overall, none of the considered embeddings outperforms the others for all the considered measures and collections. However, we observe that word2vec (Google) embeddings generally lead to the worst performances – despite the larger corpus used to train them. On the other hand, the word embeddings learned by NVSM lead to top results for MAP and P@10 in Robust04 and OHSUMED.

Table 4.12 Evaluation of DRMM using different word embeddings: word2vec (corpus), word2vec (Google), fasttext, and NVSM.

	Robust04 (t)			NY (t)		
	MAP	nDCG@100	P@10	MAP	nDCG@100	P@10
DRMM (word2vec (corpus))	0.268	**0.435**	0.455	0.141	0.211	0.274
DRMM (word2vec (Google))	0.261	0.428	0.455	0.126	0.215	0.268
DRMM (fasttext)	0.262	0.427	0.456	**0.153**	**0.226**	**0.276**
DRMM (NVSM)	**0.270**	0.434	**0.458**	0.130	0.207	0.232
	WT2g (t+d)			OHSUMED (d)		
	MAP	nDCG@100	P@10	MAP	nDCG@100	P@10
DRMM (word2vec (corpus))	0.289	0.455	0.434	0.272	**0.407**	0.375
DRMM (word2vec (Google))	**0.309**	0.455	0.460	0.238	0.361	0.360
DRMM (fasttext)	**0.309**	**0.471**	**0.468**	0.268	0.402	0.368
DRMM (NVSM)	0.294	0.458	0.442	**0.273**	0.405	**0.397**

4.6.2 Discussion

The results in Table 4.12 suggest that DRMM leans more on lexical signals rather than semantic ones. Indeed, the differences in performance are small in most cases, regardless of the word embeddings used. However, neural language models like word2vec or fasttext learn very different word representations and produce different matching scores for the same terms [182]. Thus, given the similar performances observed using different word embeddings, this implies that DRMM learns to match documents and queries by relying more on exact matching than semantic matching.

Another interesting outcome of this evaluation is the effectiveness of the word embeddings learned by NVSM. NVSM word embeddings provide top performances in Robust04 and OHSUMED – two domain-specific test collections. We recall that NVSM integrates a notion of term specificity in the learning process of word and document representations. Therefore, its word embeddings encode the co-occurrence relations between words in a collection better than neural language models. This leads to more effective matching signals that DRMM can exploit to perform retrieval. In other words, NVSM word representations are more suitable for retrieval tasks than those of neural language models.

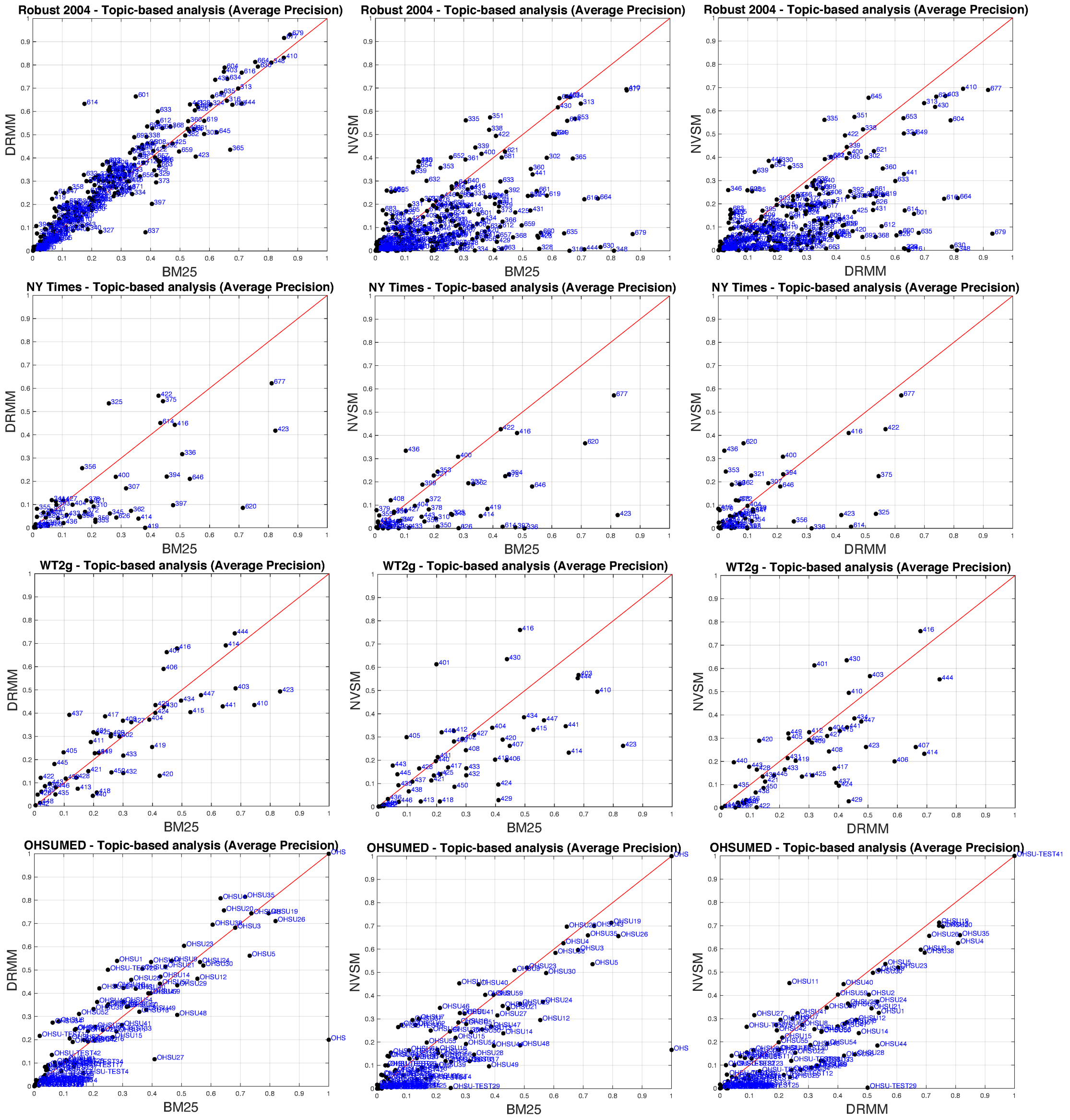
4.7 Topic-based Evaluation

We perform a topic-by-topic analysis of the rankings produced by DRMM, NVSM, and BM25. For each pair, we compare the per-topic AP@1000 obtained by the considered models. The scatter plots of the per-topic AP@1000 of each pair of models are presented in

Figure 4.4. Then, we employ KDE [237] to estimate the PDF of the AP@1000 of DRMM, NVSM, and BM25 for all the topics. Figure 4.5 shows, for each collection, the AP@1000 distributions associated to DRMM, NVSM, and BM25, while Table 4.13 reports the KLD values between the PDFs of AP@1000 for each pair of models.

4.7.1 Discussion

From Figure 4.4, we first analyze the scatter plots for Robust04. When we compare DRMM and BM25, we observe that most of the points are close to the bisector. This indicates that the two models have similar AP@1000 performances in most of the topics. On the other



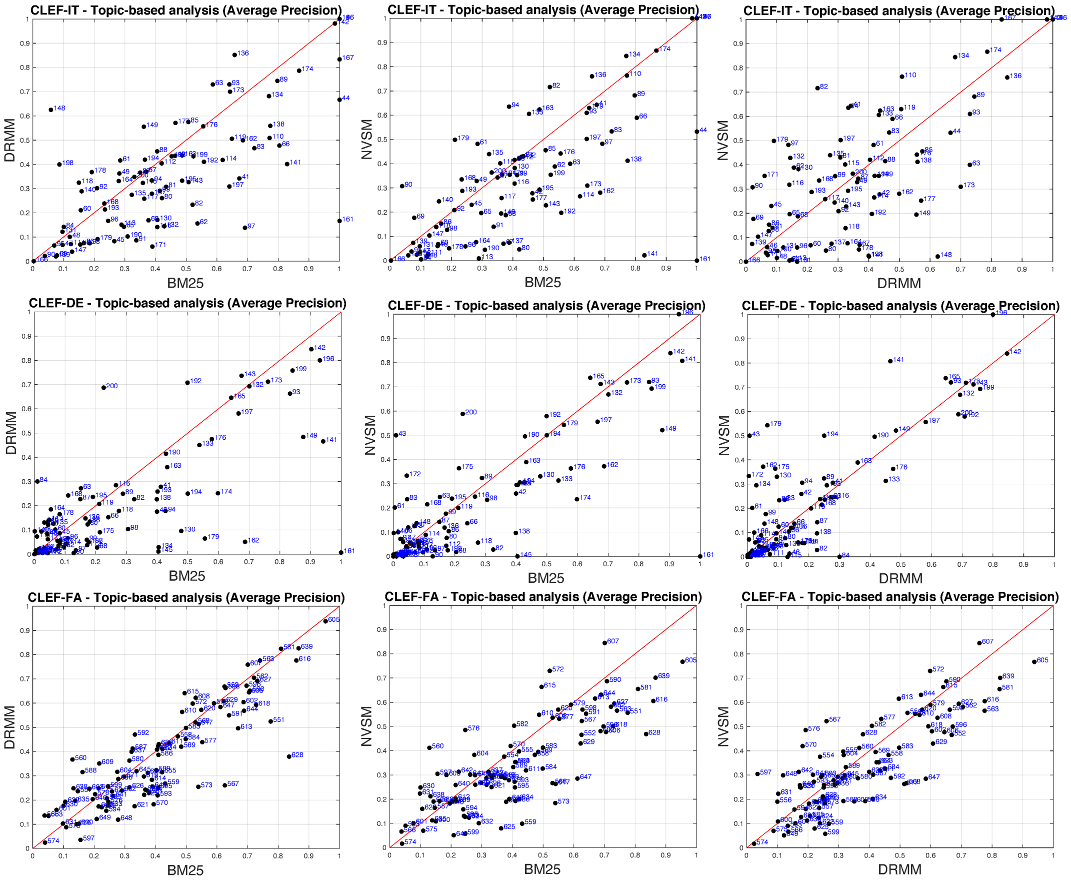


Fig. 4.4 Scatter plots of the per-topic AP@1000 scores between DRMM, NVSM, and BM25 in Robust04, NY, WT2g, OHSUMED, and CLEF collections.

hand, the comparison between NVSM and BM25 shows that most of the points are below the bisector – which indicates that BM25 outperforms NVSM in most of the topics. A similar trend also occurs between NVSM and DRMM, confirming again the tendency of DRMM to rely more on lexical matching than semantic matching.

Moving to NY, we see that both DRMM and NVSM perform worse than BM25 in most of the topics, while DRMM outperforms NVSM in a large number of topics. The poor performance of DRMM depends largely on the low number of topics in NY, which are not enough to train DRMM as effective as in other test collections. On the other hand, DRMM re-ranks a set of 2000 candidate documents retrieved by QLM (dir). Therefore, DRMM has an advantage over NVSM – which results in their performance difference. However, if we analyze the documents retrieved by NVSM and we compare them to those retrieved by BM25, we discover that NVSM retrieves relevant documents that do not contain any query term. For instance, NVSM retrieves for topic 442 ("*heroic acts*") the relevant document `1036498`, which does not contain any query term and, therefore, cannot be discovered by

BM25. In document 1036498, terms like “heroism” and “sacrifices” appear, which are highly related to the query terms. Thus, NVSM retrieves relevant documents that are most affected by the semantic gap.

As for WT2g, DRMM and BM25 behave similarly in most topics. The only notable exceptions are topics 423 and 410, where BM25 outperforms DRMM by a large margin. Compared to NVSM, both BM25 and DRMM achieve better results in most topics. However, in some topics, both DRMM and NVSM outperform BM25. An example is topic 416: “*Three Gorges Project What is the status of The Three Gorges Project?*”. In this case, the documents containing query terms are very long and BM25 fails to recognize them as relevant [152].

Similar observations can be made for OHSUMED, where DRMM performs similarly to BM25 and NVSM performs worse than both BM25 and DRMM in most topics. As in NY, NVSM retrieves relevant documents that do not contain any query term. For example, NVSM retrieves for topic OHSU7 (“lactase deficiency therapy options”) the relevant document 91359745, which does not contain any of the query terms but only synonyms or related terms (e.g., “lactose” or “intolerance”).

Regarding CLEF collections, we observe the same trend of previous collections. DRMM performs as BM25 in most topics. The only notable exceptions are topic 148 in CLEF-IT and topic 200 in CLEF-DE – where DRMM outperforms BM25 – and topics 161 and 44 in CLEF-IT, topics 141, 149, and 161 in CLEF-DE, and topic 628 in CLEF-FA – where BM25 outperforms DRMM by a large margin. On the other hand, even though BM25 outperforms NVSM in all CLEF collections for most of the topics, the difference between the two models is lower than in previous collections (e.g., Robust04 and NY). Also, the comparison between NVSM and DRMM highlights a larger difference in CLEF-IT – where the points in the scatterplot are far from the bisector – than in CLEF-DE and CLEF-FA, where there are only a few outliers in favor of NVSM in CLEF-DE.

Overall, DRMM and BM25 perform similarly in all collections (with only few outliers), while NVSM is the model that differs the most. This larger difference between DRMM and NVSM depends on the fact that DRMM re-ranks the top 2000 documents retrieved by QLM (dir) – which is a lexical model – and leans more towards lexical matching than semantic matching (see Section 4.6). On the other hand, NVSM performs retrieval over the entire document collection relying only on semantic matching. Therefore, the rankings produced by NVSM contain a more diverse set of documents than those produced by DRMM and BM25.

From the plots in Figure 4.5, we can assess the differences in models behavior that cannot be identified when analyzing average performances across topics. In this way, we discover that in Robust04, NY, CLEF-DE, WT2g, and OHSUMED, MAP values are influenced by the large number of topics where models perform poorly. Indeed, in all the plots associated

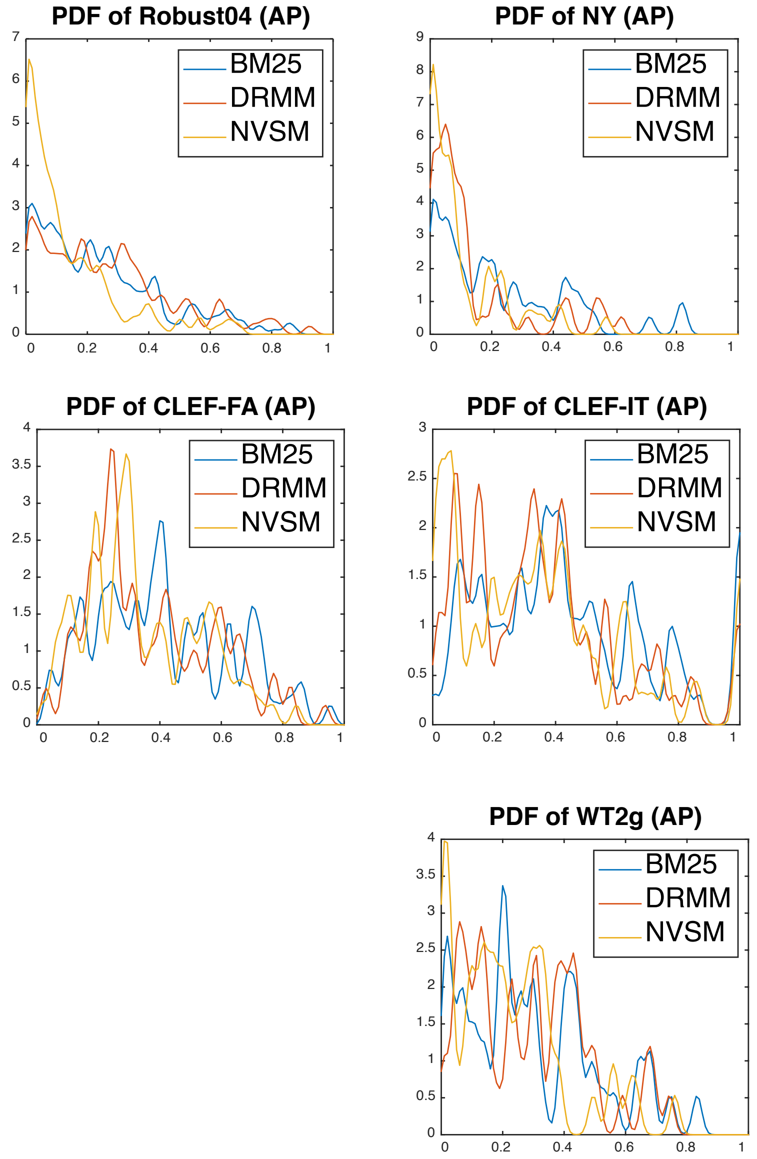


Fig. 4.5 PDFs of AP@1000 associated to DRMM, NVSM, and BM25 for Robust04, NY, WT2g, OHSUMED, and CLEF collections. Note that the *x*-axis represents the AP@1000 values distributed into 100 bins (from 0 to 1 with 0.01 step) and the *y*-axis the density estimation.

to these collections there is a high peak related to low AP1000 values. Conversely, the plots for CLEF-IT and CLEF-FA highlight that MAP values are dominated by a large number of topics where models achieve AP@1000 values around 0.3 – except NVSM, that presents a high peak also around 0.1.

Quantitatively, the distances between AP@1000 distributions can be measured with KLD. Table 4.13 reports the KLD scores between the AP@1000 distributions of DRMM, NVSM, and BM25.

Table 4.13 KLD scores between the AP@1000 PDFs of DRMM, NVSM, and BM25 in Robust04, NY, WT2g, OHSUMED, and CLEF collections. KLD $\in [0, +\infty)$ denotes the divergence between two distributions [34]. Thus, 0 means that two models behave the same way across all the topics in a collection, whereas $+\infty$ means that two models behave differently across all the topics.

	Robust04	NY	WT2g	OHSUMED
BM25 - DRMM	10.86	63.29	27.08	16.91
BM25 - NVSM	20.57	40.69	61.22	49.60
DRMM - NVSM	31.52	23.85	76.29	71.66
	CLEF-IT	CLEF-DE	CLEF-FA	
BM25 - DRMM	14.48	21.76	18.58	
BM25 - NVSM	26.65	28.15	22.70	
DRMM - NVSM	18.69	24.06	16.45	

According to the plots in Figure 4.5, the results from Table 4.13 indicate that DRMM and BM25 have the closest AP@1000 distributions in Robust04, WT2g, OHSUMED, CLEF-IT, and CLEF-DE. In plain words, DRMM and BM25 present a high number of topics where they obtain similar AP@1000 scores. However, it is worth mentioning that the topics where they achieve similar AP@1000 scores might not be the same. In fact, KLD only considers the distribution of the AP@1000 scores, ignoring the corresponding topic ids.

A different situation occurs in NY and CLEF-FA collections, where DRMM and NVSM present the closest distributions. This result further supports the high peaks associated to low AP@1000 scores – around 0.1 and 0.2 – that DRMM and NVSM share in NY and CLEF-FA, respectively.

The analysis shows that the main advantages of the considered neural IR models are the ability to retrieve relevant documents that are most affected by the semantic gap (NVSM), and the ability to effectively rank long, relevant documents that contain short, relevant passages related to a query (DRMM). On the other hand, models performances are influenced by the large number of topics where they perform poorly – in particular NVSM, that presents high peaks for lower values of AP@1000 compared to DRMM and BM25.

4.8 Chapter Outcomes and Lessons Learned

In this chapter, we investigated lexical and semantic matching signals to understand what features lexical and semantic models share, if they are complementary, and how they can be combined to effectively address the semantic gap. To this end, we evaluated the critical aspects of neural IR models through different analyses. Each analysis brought a different perspective in the understanding of semantic models and their relation with lexical models. The semantic models we selected are prominent examples of the current neural IR wave and achieve competitive performances in their field of application. The Deep Relevance Matching Model (DRMM) is a supervised approach that re-ranks candidate documents previously retrieved by a lexical (efficient) model. DRMM exploits word embeddings to extract semantic matching signals. Then, semantic matching signals are combined with lexical matching signals between query and document terms to perform re-ranking. On the other hand, the Neural Vector Space Model (NVSM) is an unsupervised approach that performs retrieval on the whole document collection. NVSM extends two unsupervised representation-based neural models for product [222] and expert [224] search to newswire retrieval. Unlike the models it extends, NVSM integrates a notion of term specificity [211, 190] in the learning process of word and document representations. For each analysis, both DRMM and NVSM have been evaluated on shared test collections – which play a fundamental role in enabling the reproducibility of the experiments.

First, we have analyzed the key factors to reproduce neural IR models. From the reproducibility study on DRMM, we highlight the importance of sharing the tools used to preprocess the document collections – that is, the tools used to prepare DRMM input data. Preprocessing is often an overlooked component of neural IR models, which has a sizable impact on the overall performances. Another key factor that needs to be shared is the training process used for word embeddings. Small changes in the training (hyper) parameters can lead to very different word embeddings, which can have a huge impact on retrieval performances. We drawn similar observations from the reproducibility study on NVSM. The lack of information about preprocessing – and especially about the creation of the word vocabulary – hampers the reproduciblity of the experiments performed by Van Gysel et al. [223].

Secondly, we have compared DRMM and NVSM with several lexical and semantic models. The lexical models considered were TF-IDF, QLM (dir), BM25, and DFR, whereas the semantic models were word2vec (add) and word2vec (si) [235]. The results of this analysis show that DRMM outperforms lexical models in Robust04. Therefore, the adoption of stronger retrieval baselines, such as BM25+RM3 [244], or the integration of contextual word embeddings [153] can further improve DRMM re-ranking effectiveness. Conversely, the poor retrieval performances obtained by DRMM in NY highlight one of the biggest bottlenecks of supervised neural IR models: the high demand for labeled data. In fact, the limited number of topics to learn from, along with the large size of the document collection, hinders the learning process of DRMM – leading to a detrimental re-ranking that worsens the initial ranking produced by QLM (dir). On the other hand, the analysis shows the effectiveness of NVSM compared to word2vec approaches in Robust04. However, the performance gap between NVSM and lexical models is still significant. A similar scenario occurs in NY, where NVSM, DRMM, and word2vec (si) achieve similar results – far from those obtained by lexical models. As for the re-ranking performed using NVSM, we see that it fails to improve the initial ranking produced by QLM (dir). By relying exclusively on the learned word and document representations to perform semantic matching, NVSM does not exploit the lexical signals provided by QLM (dir). Besides, lexical and semantic matching signals lead models to retrieve different relevant documents, as also shown in the topic-based evaluation of Section 4.7. Therefore, the unsupervised semantic nature of NVSM suits better to rank fusion or query expansion techniques – where lexical and semantic matching signals can be combined effectively (e.g., Table 4.7) – or to provide semantic features for supervised re-ranking models (e.g., Table 4.12). In other words, we believe that unsupervised semantic models should be used at the early stages of the IR pipeline rather than in re-ranking scenarios.

Thirdly, we have evaluated the impact that different domains and languages have on the performances of DRMM and NVSM. As domains, we considered Web (WT2g) and medicine (OHSUMED). For multilingualism, we considered Italian (CLEF-IT), German (CLEF-DE), and Farsi (CLEF-FA). The analysis shows the inconstancy of DRMM across different languages and its limited impact in most collections, where the only exceptions are OHSUMED and CLEF-FA. On the other hand, NVSM struggles to generalize to heterogeneous collections like WT2g. The reason is that NVSM extends two semantic models tailored to product [222] and expert [224] search, thus inheriting a domain-specific nature. For this analysis, we also performed parameter tuning over DRMM and NVSM – obtaining significant improvements in WT2g, for both DRMM and NVSM, and in CLEF-DE, for NVSM. In particular, the results of this optimization show that vocabulary size has a large impact on NVSM performances. Our intuition is that the default vocabulary size of NVSM (i.e., $|V| = 2^{16}$) is not sufficient to represent WT2g and CLEF-DE collections, which present larger vocabularies compared to the other collections considered (see Table 4.2). In fact, the increase of the NVSM vocabulary size led to performance improvements on these collections. However, improving effectiveness comes at the expense of efficiency, since a larger vocabulary size means a higher number of word representations – which, in turn, means a higher memory requirement. Therefore, finding a trade-off between NVSM vocabulary size and collection size is fundamental. As a side note, the same observations also apply to Robust04 and NY collections, which present vocabulary sizes similar to those of CLEF-DE and WT2g, respectively (cf. Tables 4.1 and 4.2).

Fourthly, we have evaluated the impact that different word embeddings have on DRMM. The word embeddings considered are: word2vec (corpus), word2vec (Google), fasttext, and NVSM. The results highlight the limited impact of different word embeddings on DRMM performances. This implies that DRMM learns to match documents and queries by relying more on lexical matching than semantic matching. Interestingly, NVSM word embeddings lead to top performances in Robust04 and OHSUMED – two domain-specific collections. Since NVSM integrates term specificity in the learning process, its word embeddings encode better co-occurrence relations and provide more effective matching signals to DRMM.

Fifthly, we have conducted an in-depth per topic analysis of the rankings produced by DRMM, NVSM, and BM25. The objective was to investigate when/where neural models fail or succeed compared to lexical models. The analysis shows that DRMM presents a behavior similar to BM25 in most collections, as opposed to NVSM. This difference in performance between DRMM and NVSM depends on two factors. First, DRMM performs re-ranking over a set of candidate documents retrieved by QLM – which is a lexical model. Secondly, DRMM exploits both lexical and semantic matching to re-rank documents and, as seen in Section 4.6, it leans more towards lexical matching. On the other hand, NVSM performs retrieval over the entire document collection relying only on semantic matching. Therefore, the rankings produced by NVSM contain a more diverse set of relevant documents than those produced by DRMM and BM25. In particular, NVSM retrieves relevant documents that are most affected by the semantic gap – that is, relevant documents that do not contain any query term. Nevertheless, this ability is not sufficient to compete with BM25 on average. This suggests that the number of semantically hard queries is limited in the considered collections. As a consequence, semantic matching has a minor impact on average performances.

The outcomes of this chapter highlighted the differences between lexical and semantic matching signals, the need to combine them at the early stages of the IR pipeline to effectively address the semantic gap, and the semantic models that are best suited for this task. The following chapters build on the insights of this chapter to develop lexical and semantic models addressing the semantic gap. In Chapter 5, we investigate how to leverage external knowledge resources to enhance the bag-of-words representations used by lexical models to perform retrieval. We focus on an important use-case in medical retrieval: providing useful precision medicine information to clinicians treating cancer patients. To this end, we develop and evaluate several knowledge-enhanced query expansion and reduction techniques. We demonstrate the effectiveness of the proposed query reformulations – especially for precision-oriented measures – and we identify a robust subset of these techniques that can be used at the early stages of the IR pipeline to effectively address the semantic gap between queries and documents. Then, we present our novel unsupervised knowledge-enhanced neural framework in Chapter 6. The framework integrates external knowledge in the learning process of neural models, and it does not require any labeled data for training. The representations learned within this framework encode linguistic features related to the semantic gap, which help to address semantically hard queries. We demonstrate the effectiveness of the models developed within this framework when used to perform retrieval over the entire document collection or to retrieve feedback documents for PRF methods – that is, when they are used at the early stages of the IR pipeline.

Chapter 5

Knowledge-Enhanced Lexical Models

The use of external knowledge to enhance query and document bag-of-words representations has a long-standing tradition in IR [229, 230, 161, 2]. Typically, knowledge-enhanced lexical models can be divided in three categories: (i) models that integrate external knowledge in the indexing stage [2], (ii) models that integrate external knowledge in the retrieval stage [230], and (iii) models that integrate external knowledge in both indexing and retrieval stages [229, 161]. In this chapter, we focus on the second category and we investigate the effectiveness of knowledge-enhanced lexical models for medical IR.

Medical IR helps a wide variety of users to access and search medical information archives and data [89]. A central issue in medical IR is the diversity of users, presenting different information needs and varying levels of medical knowledge. For example, a patient with a recently diagnosed condition would generally benefit from introductory information about the treatment of the disease, while a trained physician would require more detailed information when deciding the course of treatment.

Therefore, understanding the information needs of various users is one of the cornerstones of medical IR. In [105, Chapter 2], Hersh proposes to classify textual medical information in two categories: patient-specific information and knowledge-based information. Patient-specific information applies to individual patients and can be structured, as in the case of an EHR, or free narrative text. Knowledge-based information derives from observational or experimental research and can be organized in a wide variety of forms. In the case of clinical research, this information is most commonly provided by books and journals but can take other forms too – including computerized media.

A critical characteristic of the medical domain is the prominence of the semantic gap [70, 130, 131]. Edinger et al. [70] performed a failure analysis on the TREC Medical Track [234] to identify what impaired the retrieval systems during the track. One of the outcomes of this failure analysis highlighted that relevant documents were most often infrequently retrieved due to the use of synonyms for topic terms. Koopman and Zuccon [130] investigated if and why assessing relevance of clinical records for a clinical retrieval task is cognitively demanding. The analysis showed, among other things, that the interpretation of a considerable number of queries was subjective and often required careful consideration regarding different possible interpretations. This high degree of subjectivity to interpret queries can increase the mismatch between the machine-level description of document and query contents and their human-level interpretation. Koopman et al. [131] divided the semantic gap into core aspects, general enough to be found in any domain, and analyzed their impact in the medical one. To this end, they provided example queries where each of the considered aspects is prominent.

Along the same lines, Sondhi et al. [209] identified several challenges arising from the semantic gap in the medical domain. However, they found out that combining retrieval models with selective query term weighting, based on medical thesauri and physician feedback, proves effective to address these challenges and improves performances significantly. Similar findings were also obtained by Zhu et al. [256] and Diao et al. [62], that relied on query expansion and reweighting techniques to improve retrieval performances on medical records. Therefore, the design of effective tools to access and search textual medical information can benefit, among other things, from enhancing the query through expansion and/or rewriting techniques that leverage the information contained within external knowledge resources.

Thus, to investigate the effectiveness of knowledge-enhanced lexical models that rely on query expansion and/or rewriting techniques, we conduct a series of studies and analyses on the TREC Precision Medicine (PM) Track. From 2017 to 2019, the TREC PM Track [186, 185, 187] focused on an important use-case in Clinical Decision Support (CDS): providing useful precision medicine information to clinicians treating cancer patients. This track gives a unique opportunity to evaluate retrieval systems, as the provided test collections adopt the same set of topics – i.e., synthetic cases created by precision oncologists – for two different document sets that target two different tasks: 1) retrieving biomedical articles addressing relevant treatments for a given patient, and 2) retrieving clinical trials for which a patient is eligible.

First, we conduct a preliminary study on the TREC PM 2018 Clinical Trials task – where the objective is to retrieve relevant clinical trials for which the patient is eligible. Relevant clinical trials represent the potential for connecting patients with experimental treatments if existing treatments have been ineffective. To this end, we propose a procedure to: 1) expand queries iteratively, relying on medical knowledge resources [122, 155], to increase the probability of finding relevant trials by adding neoplastic, genetic, and proteic term variants to the original query; 2) filter out trials, based on demographic data, for which the patient is not eligible. The purpose of the study is twofold: (i) we want to evaluate how a recall-oriented approach based on an increasing – and more aggressive – query expansion method affects precision in this context; (ii) we want to investigate whether the effectiveness of the retrieval model can be correlated with the quality of the relational information contained within the knowledge resource(s) used in the expansion process. We chose the TREC PM 2018 test collection to perform this preliminary study as it is the most representative collection for precision medicine. Compared to TREC PM 2017 and 2019 test collections, TREC PM 2018 presents the largest number of topics related to cancer cases (see Subsection 2.1.3) – thus providing a more solid starting point to investigate the task.

Then, we deepen the analysis performed in the preliminary study and we extend it to both scientific literature and clinical trials retrieval. In other words, we take advantage of the dual nature of TREC PM collections and we evaluate several state-of-the-art query expansion and reduction techniques to examine whether a particular approach can be helpful in both scientific literature and clinical trials retrieval. For this analysis, we consider TREC PM 2017 and 2018 test collections and we compare our approach with the best runs submitted to the TREC PM Tracks in the two years.

Given the outcomes of the in-depth analysis, we conduct a validation study on the TREC PM 2019 Track. We focus on both tasks, with particular emphasis on the Clinical Trials task. The objective of this study is twofold: (i) we want to evaluate how the different query reformulations – tested on previous TREC PM collections – affect the results and whether the findings obtained in the previous analysis remain valid; (ii) we want to verify if combining different query reformulations based on expansion and reduction techniques proves effective in such a highly specific scenario.

Finally, we perform an a posteriori analysis on the effectiveness of the proposed query reformulations for clinical trials retrieval over the three years of TREC PM. This systematic analysis compares our approach with those proposed by the research groups that participated in all the three years of TREC PM. The experimental results show the effectiveness of the proposed query reformulations in all collections – in particular for retrieving relevant clinical trials in top positions of the ranking list.

The main contributions of this chapter are:

C1 We conduct a preliminary study on clinical trials retrieval to evaluate how a recall-oriented approach based on an increasing – and more aggressive – query expansion method affects precision in this context. The analysis of the experimental results shows that the proposed query expansion approach introduces noise and significantly decreases retrieval performances. In particular, we found that the detrimental effect of the query expansions depends on the lack of an appropriate weighting scheme on query terms and the uncontrolled use of all the knowledge resources contained within

UMLS. Thus, the study highlights what features are required to build effective query expansions, and what instead should be avoided.

C2 Based on the outcomes of the preliminary study, we propose several state-of-the-art query expansion and reduction techniques and we perform an in-depth analysis to examine whether a particular approach can be helpful in both scientific literature and clinical trials retrieval. The experimental results show that no clear pattern emerges for both tasks. Nevertheless, we found that a particular combination of query reformulations performs well in both tasks – especially for clinical trials retrieval – and achieves top performances for many evaluation measures in both TREC PM 2017 and 2018.

C3 We conduct a validation study on TREC PM 2019 to evaluate whether the effectiveness of the proposed query reformulations, demonstrated on previous TREC PM collections, still holds. The experimental results confirm the effectiveness of the tested query reformulations for clinical trials retrieval – in particular for precision-oriented measures.

The rest of this chapter is organized as follows. In Section 5.1, we describe the preliminary study we conducted on the Clinical Trials task of TREC PM 2018. In Section 5.2, we present the in-depth analysis we performed on query reformulations for scientific literature and clinical trials retrieval. In Section 5.3, we report the validation study we conducted on the tested query reformulations for TREC PM 2019. In Section 5.4, we present the a posteriori analysis we performed on the proposed query reformulations for clinical trials retrieval. Finally, in Section 5.5, we conclude the chapter with a discussion on the achieved outcomes and the lessons learned.

5.1 Preliminary Study: TREC Precision Medicine 2018

We describe the preliminary study we performed on the TREC PM 2018 Clinical Trials task. The study served to identify what features are required to build effective query expansions, and what instead should be avoided. The methodology we propose consists of a procedure to: 1) expand queries iteratively, relying on medical knowledge resources, to increase the probability of finding relevant trials by adding neoplastic, genetic, and proteic term variants to the original query; 2) filter out trials, based on demographic data, for which the patient is not eligible. The objective of the study is twofold: (i) evaluate how a recall-oriented approach based on an increasing – and more aggressive – query expansion method affects precision in this context; (ii) investigate whether the effectiveness of the retrieval model can be correlated with the quality of the relational information contained within the knowledge resource(s) used in the expansion process.

5.1.1 Methodology

The proposed methodology is composed of five steps. The steps are: indexing, pre-retrieval query expansion, retrieval, post-retrieval query expansion, and filtering.

Indexing

Indexing consists of two phases. First, we rely on MetaMap [15], a biomedical concept mapper from the NLM (see Section 3.1), to extract from each document of the collection all the concepts associated to the following UMLS semantic types: `Neoplastic Process` (*neop*) and `Gene or Genome` (*gngm*).[1] Then, we index the collection by the following fields: `<docid>`, `<text>`, `<max_age>`, `<min_age>`, `<gender>`, and `<concepts>`.
Fields `<max_age>`, `<min_age>`, and `<gender>` are extracted from the `eligibility` section of clinical trials and are required for filtering. The `<text>` field contains the entire content of each clinical trial, including the information stored within `<docid>`, `<max_age>`, `<min_age>`, and `<gender>` fields. The `<concepts>` field contains the list of UMLS CUIs extracted by MetaMap.

Pre-Retrieval Query Expansion

We perform a knowledge-based, a priori, query expansion. First, we employ MetaMap to extract from each query all the UMLS concepts belonging to the following semantic types: `Neoplastic Process` (*neop*), `Gene or Genome` (*gngm*), and `Amino Acid, Peptide, or Protein` (*aapp*). Then, for each extracted concept, we consider all its name variants. For instance, let us consider the UMLS concept “melanoma” with CUI `C0025202`. The set of name variants for “melanoma” contains: cutaneous melanoma; malignant melanoma; melanoma; melanoma malignant; mm - malignant melanoma; malignant melanomas; malignant melanoma (disorder); etc. Thus, expanded queries consist of the union of the original query terms with the set of name variants for each concept identified. In other words, extracted concepts are used as a proxy to expand the original query with highly related terms from an external knowledge resource.

[1] https://metamap.nlm.nih.gov/SemanticTypesAndGroups.shtml

Retrieval

We adopt BM25 [192] to perform retrieval. Given a collection D of N documents and a query q expressed as a bag-of-words $q = \{q_i\}_{i=1}^{n} \in V$, where q_i is a word, V is the vocabulary, and $n = |q|$ is the query length, BM25 computes the score between the query q and a document d as follows:

$$\text{score}(q,d) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{tf(q_i,d) \cdot (k_1 + 1)}{tf(q_i,d) + k_1 \cdot (1 - b + b \cdot \frac{|d|}{avgdl})} \tag{5.1}$$

where $\text{IDF}(q_i)$ is the IDF weight of the query term q_i, $tf(q_i,d)$ is the term frequency of q_i within document d, $|d|$ is the document length, k_1 and b are hyperparameters, and $avgdl$ is the average document length in the test collection from which documents are drawn. $\text{IDF}(q_i)$ is computed as:

$$\text{IDF}(q_i) = \log\left(\frac{N - N(q_i) + 0.5}{N(q_i) + 0.5}\right) \tag{5.2}$$

where N is the total number of documents in the collection and $N(q_i)$ is the number of documents that contain the query term q_i.

Post-Retrieval Query Expansion

We perform a PRF based query expansion. The set of documents retrieved by BM25 using the a priori expanded query is used to select additional expansion terms for the second round of retrieval. Given the top k retrieved documents, we select the document concepts – identified by MetaMap during the indexing step – that match the concepts associated to the query terms. Then, for each matched concept, we consider the name variants of its neighbor concepts – that is, concepts that present a hierarchical or associative relation within UMLS with the matched concept.[2] We limit the neighbor concepts to those concepts belonging to *neop*, *gngm*, and *aapp* semantic types to avoid introducing information that is not strictly related to the contents of the query. Thus, the name variants identified with the PRF based query expansion further extend the query after the a priori query expansion.

Filtering

Within clinical trials, one of the most relevant sections is the eligibility section. The eligibility section comprises, among other things, three demographic aspects that a patient needs to satisfy to be considered eligible for the trial, namely: `minimum age`, `maximum age`, and `gender`. A description of each aspect follows.

[2] https://www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/release/abbreviations.html

- `minimum age` is the minimum age required for a patient to be considered eligible for the trial, when not stated we do not set a lower bound on the eligible age of a patient.
- `maximum age` is the maximum age required for a patient to be considered eligible for the trial, when not stated we do not set an upper bound on the eligible age of a patient.
- `gender` is the gender required for a patient to be considered eligible for the trial, when not specified we consider the trial available regardless of the gender.

In order to filter our clinical trials for which a patient is not eligible, we perform a filtering step based on demographic data. Given a patient case, whenever its demographic data – i.e., the query field `<demographic>`, which contains age and gender – do not satisfy the eligibility criteria for a clinical trial, we exclude that trial from the list of candidate trials for which the patient is eligible.

5.1.2 Experimental Setup

Test Collection and Knowledge Resource

We consider the Clinical Trials collection of the TREC PM 2018 (PM18) Track [185]. We use all the query fields and we perform experiments on the 50 topics provided. As knowledge resource, we adopt the 2018AA release of the UMLS metathesaurus [29]. More details on the test collection and knowledge resource used can be found in Subsections 2.1.3 and 2.2.4, respectively.

Evaluation Measures

We use the official measures adopted in the TREC PM 2018 Track, which are infNDCG, Rprec, and P@10.

Experimental Procedure

We use Whoosh to perform indexing, retrieval, and filtering as it provides easy access and control of the functionalities required for such operations. For BM25, we keep the default values $k_1 = 1.2$ and $b = 0.75$ provided by Whoosh. We rely on MetaMap to perform concept extraction on documents and queries. We set the number of feedback documents $k = 10$ for the PRF based query expansion.

We summarize the procedure used for the experiments below.
Indexing:

- Use MetaMap to extract from each clinical trial the UMLS concepts restricted to *neop* and *gngm* semantic types.
- Index clinical trials using the following created fields: `<docid>`, `<text>`, `<max_age>`, `<min_age>`, `<gender>`, and `<concepts>`.

Pre-Retrieval Query Expansion:

- Use MetaMap to extract from each query field the UMLS concepts restricted to the following semantic types: *neop* for `<disease>`, *gngm* and *comd* for `<gene>`.
- Obtain from extracted concepts all the name variants belonging to the knowledge sources contained within UMLS.
- Expand queries with the name variants of extracted concepts.

First Round of Retrieval:

- Perform a search using a priori expanded queries with BM25.

Post-Retrieval Query Expansion:

- Take the top *k* clinical trials retrieved by BM25 using the a priori expanded query.
- Select document concepts that match the concepts associated to query terms.
- Select neighbor concepts – restricted to *neop*, *gngm*, and *aapp* semantic types – that present a hierarchical or associative relation within UMLS with matched concepts.
- Obtain from neighbor concepts all the name variants belonging to the knowledge sources contained within UMLS.
- Expand the a priori expanded query with the name variants of neighbor concepts.

Second Round of Retrieval:

- Perform a search using PRF based expanded queries with BM25.

Filtering:

- Filter out candidate clinical trials for which the patient is not eligible.

We consider three different combinations of the above procedure to address the objectives of our study. The first combination (base) performs indexing, retrieval, and filtering – that is, pre- and post-retrieval query expansions are not applied. The second combination (QE) adds the pre-retrieval query expansion to the pipeline, whereas the third one (QE/PRF) performs all the steps described above. In this way, we can evaluate how increasing – and more aggressive – query expansions affect the results and whether retrieval performances can be correlated with relational information within UMLS.

5.1.3 Experimental Results

The organizers of the TREC PM 2018 Track provided the summary of the results in terms of best, median, and worst value for each topic for P@10, infNDCG, and Rprec. In Table 5.1, we report the results of the three considered models – that is, base, QE, and QE/PRF – for the three evaluation measures averaged across topics, as well as the median values for the Clinical Trials task.

Table 5.1 Retrieval performances of the considered models on the TREC PM 2018 Clinical Trials task. Median refers to the average median values of the Clinical Trials task and it is computed considering all the runs submitted to the task. **Bold** values represent the highest scores among models and median.

	P@10	infNDCG	Rprec
base	**0.5680**	**0.5421**	**0.4142**
QE	0.2920	0.3003	0.1908
QE/PRF	0.1180	0.1468	0.0865
median	0.4680	0.4297	0.3268

The results from Table 5.1 show that BM25 performs best when none of the developed query expansions are used. On average, the base model outperforms median values by a large margin – with an average gap grater than or equal to 0.10 for all measures. On the other hand, the use of both pre- and post-retrieval query expansions significantly worsens performances. QE and QE/PRF models achieve scores lower than the median values for all measures. In particular, QE/PRF shows the lowest performances among the three models considered. This suggests that the proposed knowledge-enhanced models are sensitive to topic drift – which often occurs when the query is expanded with terms that are not pertinent to the information need [228].

In order to better understand the performances of the proposed models, we perform a per-topic analysis that compares, for each measure, the three models with the task median values. Figures 5.1–5.3 display, topic by topic, the difference in performance between each model and the median values. For a positive difference (model better than median), a green barplot is shown, whereas for a negative difference (model worse than median), a red barplot is shown.

The analysis of Figures 5.1–5.3 confirms the trend found for average performances. The base model performs consistently better than task median values for most queries. Conversely, both pre- and post-retrieval expansions significantly worsen the performances for most queries. We attribute this performance drop to two main reasons: (i) we do not apply a weighting scheme on the query terms, (ii) we employ all the knowledge resources contained within UMLS. Applying a weighting scheme on query terms can reduce the impact of noisy terms when performing retrieval, while selecting a specific subset of knowledge resources can improve the quality of the extracted concepts and terms. For this reason, in Section 5.2, we perform an in-depth analysis of pre-retrieval query reformulation techniques to understand if the use of weighting schemes and tailored knowledge resources can be beneficial for retrieval effectiveness. Besides, improving the effectiveness of pre-retrieval techniques has a positive effect also on post-retrieval ones, like PRF, as the number of relevant documents retrieved in the first round grows larger.

Comparison with TREC PM 2018 Top Systems

When we look at the detailed analysis in the TREC PM 2018 overview [185], we see that the base model is one of the top 10 performing systems for all the evaluation measures in the Clinical Trials task. Specifically, it is the second best system for P@10 and Rprec, and the third one for infNDCG. Besides, the performance variations of our model across topics are among the smallest ones of all top-performing systems (see "IMS_TERM" run in Figure 3 of TREC PM 2018 overview [185]). This is a promising result, as it highlights the robustness of our baseline model to the set of topics considered. Thus, in the next section, we keep this model as a core component to investigate the effectiveness of pre-retrieval query reformulations relying on weighting schemes and specific knowledge resources.

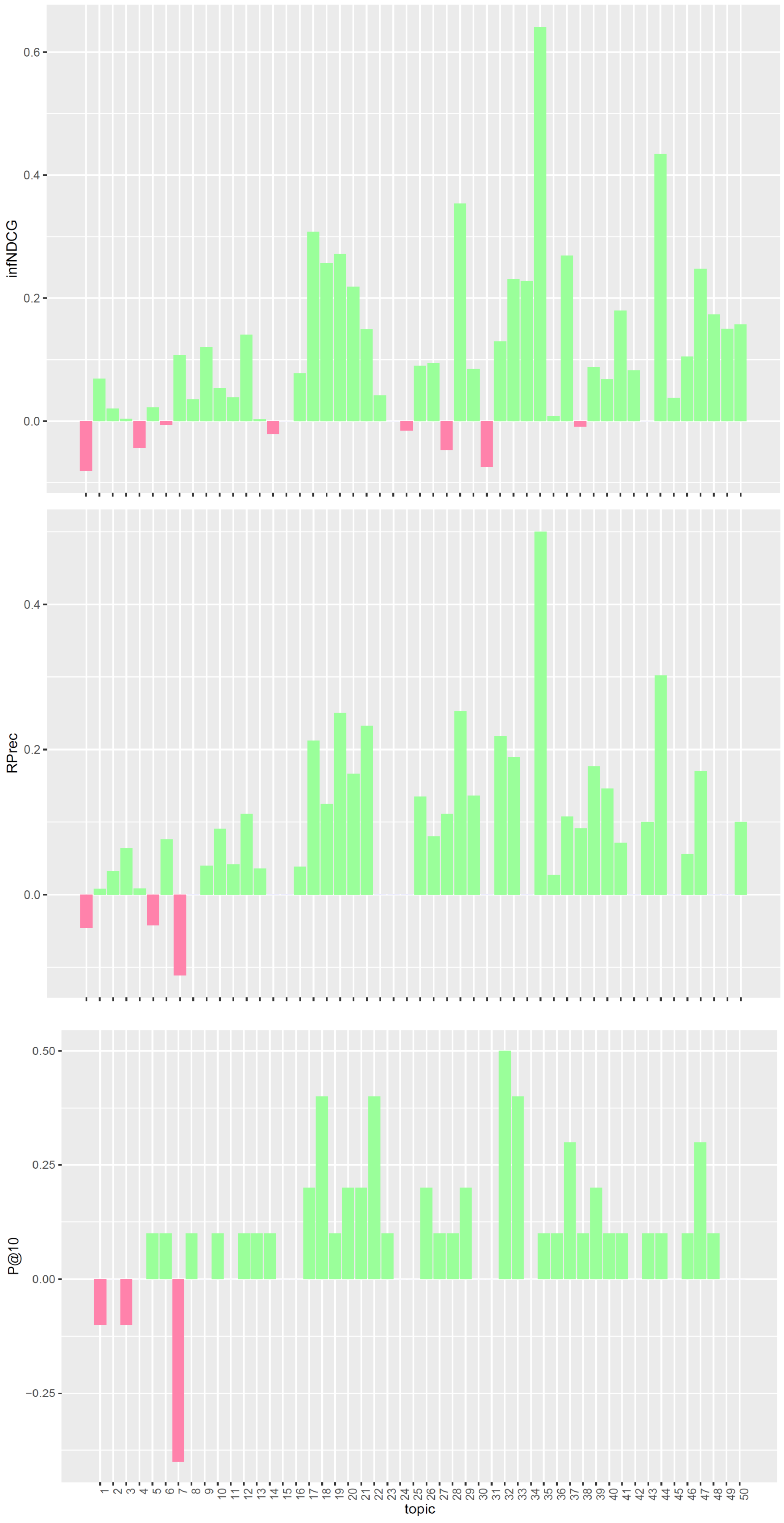


Fig. 5.1 Per-topic difference between base model and clinical trials median values. For a positive difference (model better than median), a green barplot is shown, whereas for a negative difference (model worse than median), a red barplot is shown.

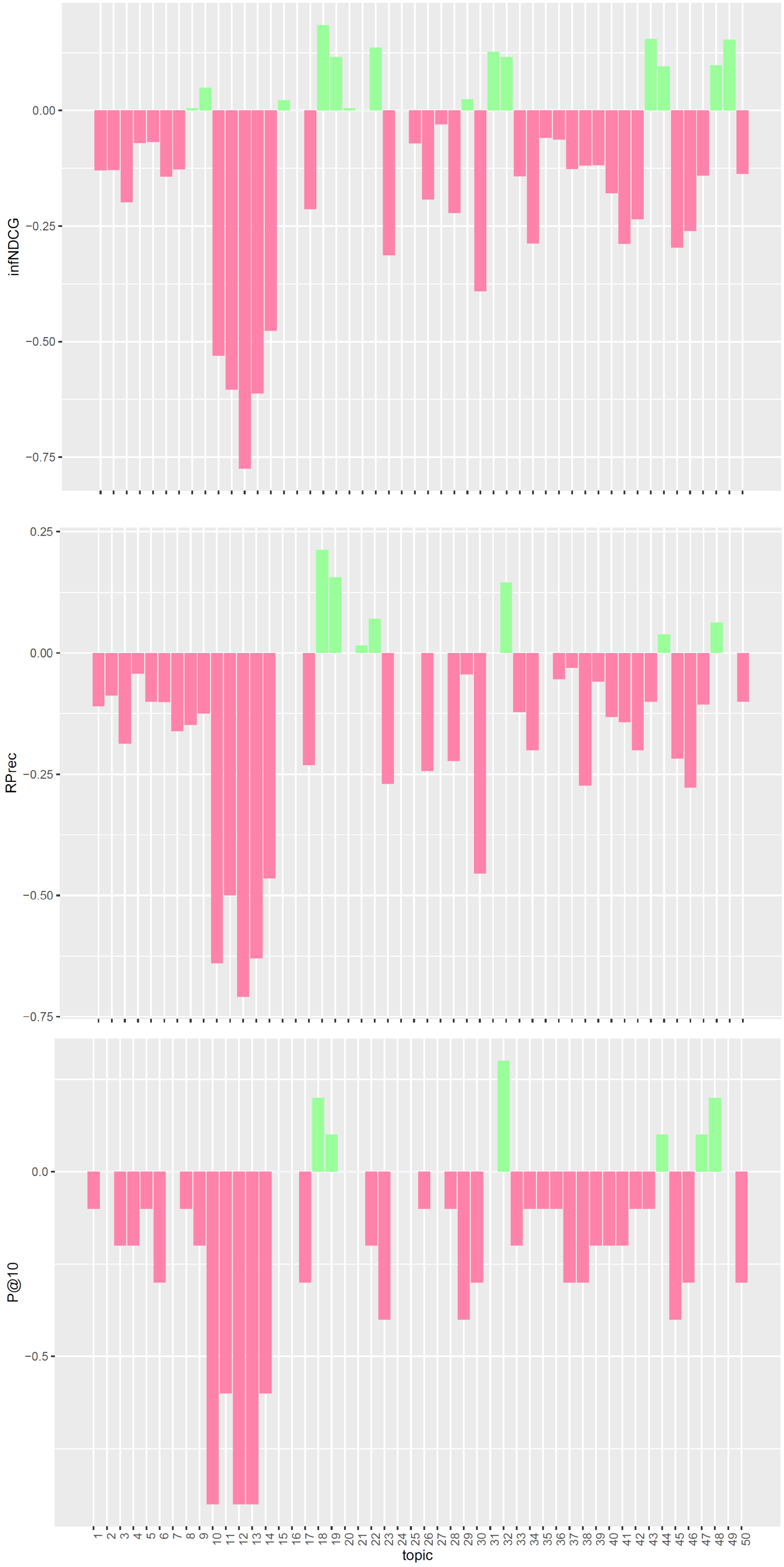


Fig. 5.2 Per-topic difference between QE model and clinical trials median values. For a positive difference (model better than median), a green barplot is shown, whereas for a negative difference (model worse than median), a red barplot is shown.

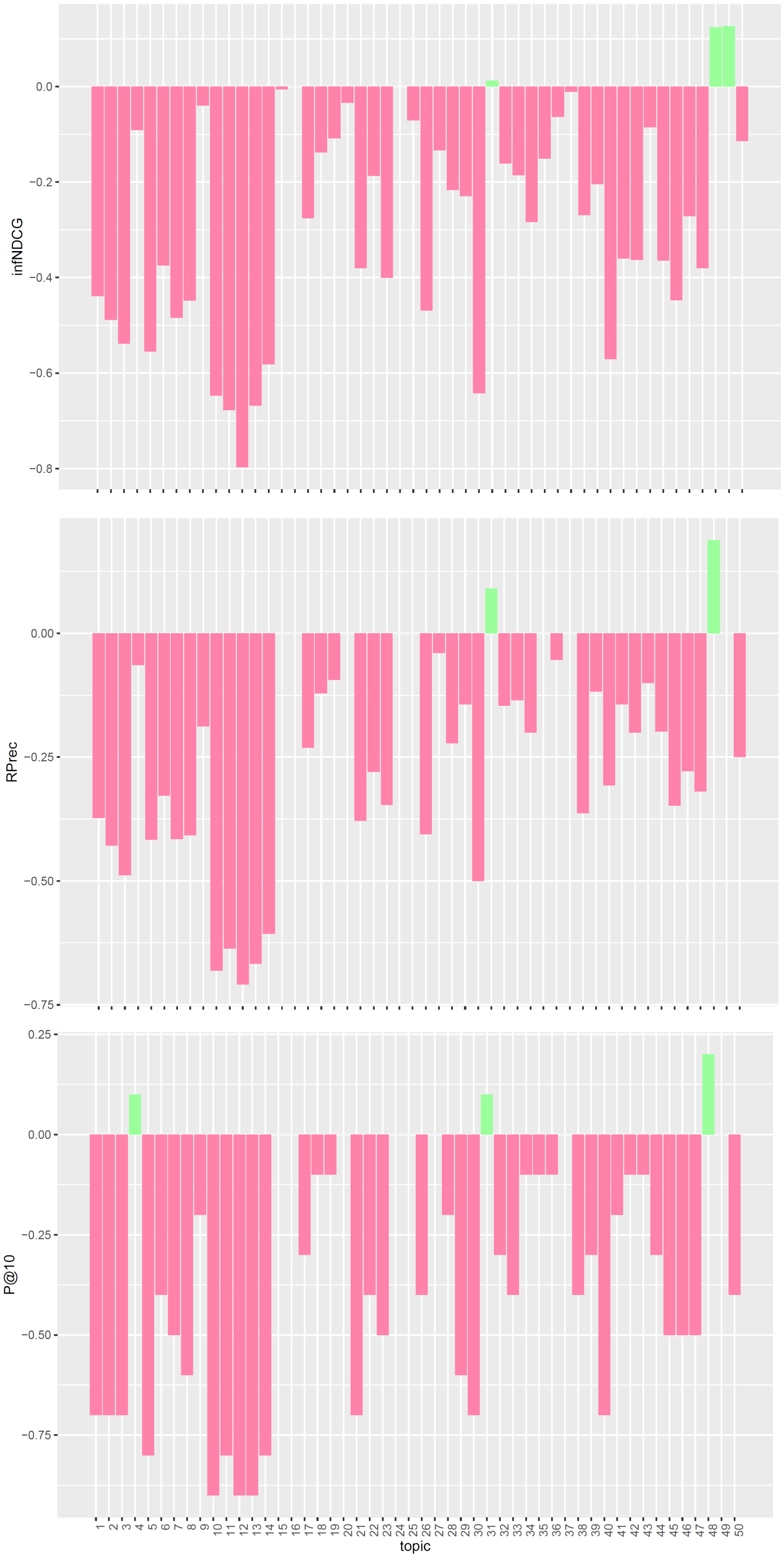


Fig. 5.3 Per-topic difference between QE/PRF model and clinical trials median values. For a positive difference (model better than median), a green barplot is shown, whereas for a negative difference (model worse than median), a red barplot is shown.

5.2 In-Depth Analysis of Query Reformulations

Given the outcomes of the preliminary study, we perform an in-depth analysis of pre-retrieval query reformulations. Compared to the previous study, we perform the analysis on both tasks of TREC Precision Medicine – that is, Scientific Literature and Clinical Trials – and we consider the test collections from 2017 and 2018 tracks. In this way, we leverage the dual nature of TREC PM collections, and we evaluate several pre-retrieval query expansion and reduction techniques to investigate whether a particular combination can be helpful in both scientific literature and clinical trials retrieval.

5.2.1 Methodology

The proposed methodology comprises three steps, plus an additional fourth step required only for the retrieval of clinical trials. The steps are: indexing, query reformulation, retrieval, and filtering.

Indexing

We use the following fields to index clinical trials collections: `<docid>`, `<text>`, `<max_age>`, `<min_age>`, and `<gender>`. Fields `<max_age>`, `<min_age>`, and `<gender>` contain information extracted from the `eligibility` section of clinical trials and are required for the filtering step. The `<text>` field contains the entire content of each clinical trial – and therefore also the information stored within the fields described above.

On the other hand, we rely on the following fields to index scientific literature collections: `<docid>` and `<text>`. As for clinical trials, the `<text>` field contains the entire content of each target document.

Query Reformulation

The proposed approach relies on two types of query reformulation techniques: query expansion and query reduction.

Query expansion: We perform a knowledge-based query expansion. First, we rely on MetaMap [15] to extract from each query field all the UMLS concepts belonging to the following semantic types: `Neoplastic Process` (*neop*), `Gene or Genome` (*gngm*) and `Cell or Molecular Dysfunction` (*comd*). The *gngm* and *comd* semantic types are related to the query `<gene>` field, while *neop* is related to the `<disease>` field. Also, for those collections where an additional `<other>` field is included – which considers other potential factors that may be relevant – MetaMap is used on <other> with no restriction on the semantic types, as its content does not consistently refer to any particular semantic type. Secondly, for each extracted concept, we consider all its name variants contained within the following knowledge resources: NCI thesaurus [205], MeSH thesaurus [146], SNOMED CT [66], and UMLS metathesaurus [29]. All the knowledge resources are authoritative and manually curated by professionals. Finally, expanded queries consist in the union of the original terms with the set of name variants.

Additionally, we expand queries that do not mention any kind of blood cancer (e.g., "lymphoma" or "leukemia") with the term "solid". This expansion proved to be effective in [90], where the authors found that a large part of relevant clinical trials do not mention the exact topic disease. In this case, a general term like "solid tumor" was preferable and more effective for retrieving relevant clinical trials.

Query reduction: We reduce queries by removing, whenever present, gene mutations from the <gene> field. We rely on the Cancer Biomarkers database [219] to identify gene mutations. To clarify, let us consider a topic where the <gene> field mentions "BRAF (V600E)". First, we verify that "BRAF (V600E)" exists within the Cancer Biomarkers database. Then, we remove the mutation "(V600E)" from the query. After the reduction process, the <gene> field becomes "BRAF". The reduction process aims to mitigate the over-specificity of topics, as the information contained within topics can be too specific compared to that contained within target documents [172].

Additionally, we remove the <other> field from those collections that include it, as it contains additional factors that are not necessarily relevant – thus representing a potential source of noise in retrieving precise information for patients.

Retrieval

We adopt BM25 [192] to perform retrieval. Furthermore, we weight the expansion terms less than 1.0 to limit noise injection in the retrieval process [97].

Filtering

After retrieval, we filter out clinical trials for which a patient is not eligible. We perform filtering based on demographic data. In those cases where part of the demographic data is not specified, a clinical trial is kept or discarded based on the remaining demographic information. For instance, if the clinical trial does not specify a required minimum age, then it is kept or discarded based on maximum age and gender values.

5.2.2 Experimental Setup

Test Collections and Knowledge Resources

We consider TREC PM 2017 (PM17) [186] and 2018 (PM18) [185] Tracks. We perform experiments on both Scientific Literature and Clinical Trials collections using the 30 and 50 topics provided, respectively, in 2017 and 2018. More details on the test collections used can be found in Subsection 2.1.3. For query expansion, we adopt the following knowledge resources: NCI thesaurus [205], MeSH thesaurus [146], SNOMED CT [66], and UMLS metathesaurus [29]. For each resource, we consider the version contained within the 2018AA release of UMLS. For query reduction, we rely on the Cancer Biomarkers database [219]. In this case, we adopt the database version of January 17, 2018. A detailed description of the considered knowledge resources can be found in Section 2.2.

Evaluation Measures

We use the official measures adopted in the TREC PM Tracks, which are infNDCG, Rprec, and P@10. We do not compute infNDCG for the 2017 Clinical Trials task, since the sampled relevance judgments are not available. On the other hand, we do not report P@5 and P@15 since they were used only for the 2017 Clinical Trials task and then replaced by infNDCG and Rprec in 2018 and 2019.

Experimental Procedure

We use Whoosh for indexing, retrieval, and filtering. For BM25, we keep the default values $k_1 = 1.2$ and $b = 0.75$ provided by Whoosh. For query expansion, we rely on MetaMap to extract and disambiguate UMLS concepts.

We summarize the procedure used for each experiment below.
Indexing:

- Index clinical trials using the following created fields: `<docid>`, `<text>`, `<max_age>`, `<min_age>`, and `<gender>`.
- Index scientific literature using the following created fields: `<docid>` and `<text>`.

Query Reformulation:

- Use MetaMap to extract from each query field the UMLS concepts restricted to the following semantic types: *neop* for `<disease>`, *gngm*/*comd* for `<gene>`, and *all* for `<other>`.

- Obtain from extracted concepts all the name variants belonging to NCI, MeSH, SNOMED CT, and UMLS metathesaurus knowledge resources.
- Expand (or not) topics that do not mention any kind of blood cancer with the term "solid".
- Reduce (or not) queries by removing, whenever present, gene mutations from the `<gene>` field.
- Remove (or not) the `<other>` field from those collections that include it.

Retrieval:

- Adopt any combination of the previous reformulation strategies.
- Weight expanded terms with a value $m \in \{0.0, 0.1, 0.2, ..., 1.0\}$.
- Perform a search using reformulated queries with BM25.

Filtering:

- Filter out candidate clinical trials for which the patient is not eligible.

5.2.3 Experimental Results

In Table 5.2, we report the results of our experiments (upper part) and compare them with the top-performing systems at TREC PM 2017 and 2018 (lower part). For each year and for each task, we present our top 5 query reformulations ordered by P@10. Each line shows a particular combination (*y*es or *n*o values) of semantic types (*neop*, *comd*, *gngm*), usage and expansion of `<other>` field (oth, oth_exp), query reduction (orig), and expansion using weighted "solid" (tumor) keyword. We report the results for both Scientific Literature (sl) and Clinical Trials (ct) tasks. We highlight in **bold** the top 3 scores for each measure, and we use the symbols † and ‡ to indicate two combinations that perform well in both years. Regarding TREC PM systems, we select systems from those participants who submitted runs in both years and reached top 10 performances for at least two measures [186, 185]. The results reported in the lower part of Table 5.2 indicate the best score obtained by a particular system for a specific measure; in general, the best results of a participant's system are often related to different runs. The symbol '$-$' means that the measure is not available, while '$<$' indicates that none of the runs submitted by the participant achieved top 10 performances. For comparison, we add for each measure the lowest score required to enter the top 10 TREC results list, and the score obtained by our best combination. The combination is indicated by the line number, which refers to its position in the upper part of Table 5.2.

Analysis of Query Reformulations

The results from Table 5.2 (upper part) highlight different trends in 2018 and 2017. In 2018, there is a clear distinction in terms of performances among the combinations that achieve the best results for Scientific Literature and Clinical Trials tasks. For Scientific Literature, considering the semantic type *neop* expansion without using the umbrella term "solid" provides the best performances for all the measures considered. On the other hand, two of the best three runs for Clinical Trials (lines 5 and 9) use no semantic type expansion, but rely on the "solid" (tumor) expansion with weight 0.1.

In 2017, the situation is completely different. Lines 12 and 13 show two combinations that achieve top 3 performances for both Scientific Literature and Clinical Trials tasks. These two combinations use query reduction and a weighted 0.1 "solid" (tumor) expansion. The use of a weighted 0.1 "solid" expansion, as well as a reduced query (orig = *n*), seems to improve performances consistently for all measures in 2017. The semantic type *gngm* seems more effective than *neop*, while *comd* does not seem to have a positive effect at all.

Thus, the analysis of query reformulations shows that no clear pattern emerges for both tasks. Overall, a query expansion approach using a selected set of semantic types helps the retrieval of scientific literature. On the other hand, a query reduction approach and a "solid" (tumor) expansion improve performances on clinical trials retrieval. Nevertheless, most of the proposed query reformulations perform well for both tasks. Besides, we found that a particular combination (marked as ‡ in Table 5.2) could have been one of the top 10 performing runs for many evaluation measures in both TREC PM 2017 and 2018.

Comparison with TREC PM Systems

The results from Table 5.2 (lower part) mark a clear distinction between the performances of participants' systems for Scientific Literature and Clinical Trials tasks. For Scientific Literature, many of the participants' runs do not reach the top 10 threshold for the considered measures – especially in 2017. The only exceptions are the systems of the research group from the University of Delaware (see udel_fang in Table 5.2), whose best runs always achieve top 10 performances for this task. Conversely, most of the participants' runs achieve top 10 performances for Clinical Trials. In particular, all participants' runs surpass the top 10 threshold in 2018.

When we consider the top 3 query reformulations from Table 5.2 (upper part), we see that they achieve performances higher than the top 10 threshold for most measures. The only exception is P@10 in the 2018 Scientific Literature task, where none of the three best query reformulations reaches the top 10 threshold. Compared to the participants' systems, most of our query reformulations achieve higher performances for most measures in both the 2017 and 2018 Clinical Trials tasks. The only notable exception is P@10 in 2017, where the system of the research group from the University of Texas at Dallas (see UTDHLTRI in Table 5.2) outperforms our top 5 query reformulations. A different situation occurs for Scientific Literature tasks, where our query reformulations do not achieve best performances for any measure in both 2017 and 2018. Nevertheless, the top 3 query reformulations achieve better results than most of the considered participants' systems.

Thus, the in-depth analysis, which stemmed from our preliminary study on the TREC PM 2018 Track, shows the effectiveness of applying a weighting scheme on expansion terms and selecting tailored knowledge resources for query expansion and reduction techniques. In particular, the results highlight the robustness of our approach across different collections and tasks. Therefore, in the next section, we conduct a validation study on the TREC PM 2019 Track to investigate whether the findings of this analysis remain valid. In other words, we evaluate how the proposed query reformulations generalize to TREC PM 2019 collections, how they affect retrieval performances, and if the trends found are confirmed.

Table 5.2 Results for the TREC PM 2017 and 2018 Tracks. Upper part reports the results achieved using the five most effective query reformulations for each year. (†) and (‡) indicate two particular combinations effective in both years. Lower part reports the results obtained by participants' systems, the lowest score required to enter the top 10 TREC results list, and the score obtained by our best combination. Further details are reported in Subsection 5.2.3.

		Semantic Type			Field Other				sl	ct	sl	ct	sl	ct
line	year	neop	comd	gngm	oth	oth_exp	orig	solid	P_10	P_10	infNDCG	infNDCG	Rprec	Rprec
1	2018	y	y	n	n	n	y	n	**0.5660**	0.5540	**0.4912**	0.5266	**0.3288**	0.4098
2	2018	y	n	n	n	n	y	n	**0.5640**	0.5600	**0.4961**	0.5264	**0.3288**	0.4138
3	2018	y	n	y	n	n	y	n	**0.5480**	0.5660	**0.4941**	0.5292	**0.3266**	0.4116
4	2018	n	n	n	n	n	y	n	0.5460	0.5680	0.4876	**0.5411**	0.3240	**0.4197**
5	2018	n	n	n	n	n	y	0.1	0.5440	**0.5740**	0.4877	**0.5403**	0.3247	**0.4179**
6	2018	n	y	n	n	n	y	n	0.5440	0.5540	0.4853	**0.5403**	0.3236	0.4130
7	2018	y	n	n	n	n	n	n	0.5420	**0.5700**	0.4636	0.5345	0.3180	0.4134
8†	2018	n	n	y	n	n	y	n	0.5340	0.5640	0.4877	0.5337	0.3229	0.4106
9‡	2018	n	n	n	n	n	n	0.1	0.5300	**0.5820**	0.4635	**0.5446**	0.3148	**0.4205**
10	2018	y	n	y	n	n	n	n	0.5140	0.5680	0.4572	0.5393	0.3144	0.4122
11	2017	y	n	y	n	n	n	0.1	**0.5033**	0.3759	**0.3984**	-	0.2697	0.3206
12	2017	n	n	y	n	n	n	0.1	**0.4900**	**0.3931**	0.3881	-	0.2677	**0.3263**
13‡	2017	n	n	n	n	n	n	0.1	**0.4800**	**0.4034**	0.3931	-	**0.2728**	**0.3361**
14	2017	y	n	n	n	n	n	0.1	0.4767	0.3862	**0.3974**	-	**0.2714**	0.3202
15	2017	n	n	n	n	n	n	n	0.4733	**0.3931**	**0.3943**	-	**0.2732**	0.3241
16	2017	y	n	y	n	n	y	0.1	0.4733	0.3828	0.3567	-	0.2329	**0.3253**
17†	2017	n	n	y	n	n	y	n	0.4633	0.3862	0.3442	-	0.2254	0.3243
		TREC PM Participant Identifier												
18	2018	UTDHLTRI							0.6160	0.5380	0.4797	0.4794	<	0.3920
19	2018	UCAS							0.5980	0.5460	0.5580	0.5347	0.3654	0.4005
20	2018	udel_fang							0.5800	0.5240	0.5081	0.5057	0.3289	0.3967
21	2018	NOVASearch							<	0.5520	<	0.4992	<	0.3931
22	2018	Poznan							<	0.5580	<	0.4894	<	0.4101
	2018	Top 10 threshold							0.5800	0.5240	0.4710	0.4736	0.2992	0.3658
	2018	Best combination of our approach							(1) 0.5660	(9‡) 0.5820	(2) 0.4961	(9‡) 0.5446	(1) 0.3288	(9‡) 0.4205
23	2017	UTDHLTRI							0.6300	0.4172	0.4647	-	0.2993	-
24	2017	udel_fang							0.5067	<	0.3897	-	0.2503	-
25	2017	NOVASearch							<	0.3966	<	-	<	-
26	2017	Poznan							<	0.3690	<	-	<	-
27	2017	UCAS							<	0.3724	<	-	0.2282	-
	2017	Top 10 threshold							0.4667	0.3586	0.3555	-	0.2282	-
	2017	Best combination of our approach							(11) 0.5033	(13‡) 0.4034	(11) 0.3984	-	(15) 0.2732	(13‡) 0.3361

5.3 Validation Study: TREC Precision Medicine 2019

Given the outcomes of the in-depth analysis of query reformulations, we conduct a validation study on the TREC PM 2019 Track. We perform experiments on both tasks, with a particular focus on the Clinical Trials task. The objective of the study is twofold. First, we want to validate the effectiveness of the top query reformulations found in the previous analysis for the 2019 track. Secondly, we want to verify if combining the rankings obtained using such query reformulations proves effective.

5.3.1 Methodology

The proposed methodology comprises four steps, plus an additional step used only for retrieving clinical trials. The steps are: indexing, query reformulation, retrieval, and filtering (only for clinical trials). Then, the rankings obtained with different query reformulations are combined using rank fusion.

Indexing

We rely on the following fields to index clinical trials: `<docid>`, `<text>`, `<max_age>`, `<min_age>`, and `<gender>`. On the other hand, we use the following fields to index scientific literature: `<docid>` and `<text>`.

Query Reformulation

The approach leverages two types of query reformulation techniques: query expansion and query reduction.

Query expansion: We perform a knowledge-based query expansion. We use MetaMap [15] to extract and disambiguate from each query field all the UMLS concepts belonging to the following semantic types: `Neoplastic Process` (*neop*), `Gene or Genome` (*gngm*) and `Cell or Molecular Dysfunction` (*comd*). The *gngm* and *comd* semantic types are related to the query `<gene>` field, whereas *neop* is related to the `<disease>` field. Then, for each extracted concept, we consider all its name variants contained into the following knowledge resources: NCI thesaurus [205], MeSH thesaurus [146], SNOMED CT [66], and UMLS metathesaurus [29]. Expanded queries consist in the union of the original terms with the set of name variants.

Additionally, we expand queries that do not mention any kind of blood cancer with the term “solid”.

Query reduction: We reduce queries by removing, whenever present, gene mutations from the <gene> field. We rely on the Cancer Biomarkers database [219] to identify gene mutations.

Retrieval

We rely on BM25 [192] to retrieve documents. Query terms obtained through query expansion are weighted less than 1.0 to avoid noise injection in the retrieval process [97].

Filtering

We filter out from the list of candidate trials those for which a patient is not eligible. We perform filtering based on demographic data. In those cases where part of the demographic data are not specified, a clinical trial is kept or discarded on the basis of the remaining demographic information.

Rank Fusion

We perform rank fusion over the rankings obtained with the three most effective query reformulations for Clinical Trials and Scientific Literature tasks. We select query reformulations based on P@10, and we prioritize reformulations that are effective in the 2018 track since it shows more commonalities with the 2019 track compared to the 2017 track. We adopt CombSUM [200] to perform rank fusion and we normalize scores using min-max normalization.

5.3.2 Experimental Setup

Test Collection and Knowledge Resources

We consider the TREC PM 2019 (PM19) Track [187]. We perform experiments on both Scientific Literature and Clinical Trials collections using the 40 topics provided. The reader can find more details on the test collections used in Subsection 2.1.3. For query expansion, we adopt the following knowledge resources: NCI thesaurus [205], MeSH thesaurus [146], SNOMED CT [66], and UMLS metathesaurus [29]. For each resource, we consider the version contained within the 2018AA release of UMLS. For query reduction, we rely on the Cancer Biomarkers database [219]. We adopt the database version of January 17, 2018.

Evaluation Measures

We use the official measures adopted in the TREC PM 2019 Track, which are infNDCG, Rprec, and P@10.

Experimental Procedure

We use two different search engine libraries to index, retrieve, and filter the given collections: Whoosh for Clinical Trials and ElasticSearch for Scientific Literature.[3] We moved from Whoosh to ElasticSearch for Scientific Literature because Whoosh could not efficiently handle the increased collection size – which presents over two million documents more than the 2017 and 2018 collections (cf. Scientific Literature collections in Subsection 2.1.3). For BM25, we keep default values $k_1 = 1.2$ and $b = 0.75$ – as we found them to be a good combination for the considered tasks (cf. Sections 5.1 and 5.2). For query expansion, we rely on MetaMap to extract and disambiguate UMLS concepts.

We summarize the procedure used for each experiment below.

Indexing:

- Index clinical trials using the following created fields: `<docid>`, `<text>`, `<max_age>`, `<min_age>` and `<gender>`.
- Index scientific literature using the following created fields: `<docid>` and `<text>`.

Query Reformulation:

- Use MetaMap to extract from each query field the UMLS concepts restricted to the following semantic types: *neop* for `<disease>`, *gngm*/*comd* for `<gene>`.
- Obtain from extracted concepts all the name variants belonging to NCI, MeSH, SNOMED CT, and UMLS metathesaurus knowledge resources.
- Expand (or not) topics that do not mention any kind of blood cancer with the term “solid”.
- Reduce (or not) queries by removing, whenever present, gene mutations from the `<gene>` field.

Retrieval:

- Adopt the three most effective query reformulations from Section 5.2.

[3] https://www.elastic.co/elasticsearch/

- Weight expanded terms with $m = 0.1$.
- Perform a search using reformulated queries with BM25.

Filtering:

- Filter out clinical trials for which the patient is not eligible.

Rank Fusion:

- Perform rank fusion using CombSUM and Min Max normalization over the three most effective query reformulations for Clinical Trials.
- Perform rank fusion using CombSUM and Min Max normalization over the three most effective query reformulations for Scientific Literature.

To address the objectives of our study, we consider five different combinations of the above procedure for each task.

Clinical Trials:

- base: refers to the baseline model, that is BM25 plus filtering.
- neop/reduced: refers to *neop* expansion over reduced queries.
- solid/original: refers to “solid” expansion over original queries.
- solid/reduced: refers to “solid” expansion over reduced queries.
- qrefs/combined: refers to the combination of the above query reformulations using CombSUM.

Scientific Literature:

- base: refers to the baseline model, that is BM25.
- neop/original: refers to *neop* expansion over original queries.
- neop+comd/original: refers to *neop* and *comd* expansions over original queries.
- neop+gngm/original: refers to the *neop* and *gngm* expansions over original queries.
- qrefs/combined: refers to the combination of the above query reformulations using CombSUM.

5.3.3 Experimental Results

The organizers of the TREC PM 2019 Track provided the summary of the results in terms of best, median, and worst value for each topic for P@10, infNDCG, and Rprec. In Tables 5.3 and 5.4, we report the results of the five considered models, as well as the median values, for the Clinical Trials and Scientific Literature tasks, respectively.

Table 5.3 Retrieval performances of the considered models on the TREC PM 2019 Clinical Trials task. Median refers to the average median values of the Clinical Trials task and it is computed considering all the runs submitted to the task. **Bold** values represent the highest scores among models and median.

	P@10	infNDCG	Rprec
base	0.5053	0.6186	0.4337
neop/reduced	0.5237	0.5755	0.4135
solid/original	**0.5368**	**0.6239**	**0.4386**
solid/reduced	0.5316	0.5940	0.4264
qrefs/combined	0.5342	0.5706	0.4381
median	0.4658	0.5137	0.3477

Table 5.4 Retrieval performances of the considered models on the TREC PM 2019 Scientific Literature task. Median refers to the average median values of the Scientific Literature task and it is computed considering all the runs submitted to the task. **Bold** values represent the highest scores among models and median.

	P@10	infNDCG	Rprec
base	0.5125	**0.4747**	0.2977
neop/original	0.5150	0.4645	0.2982
neop+comd/original	0.5125	0.4636	0.2964
neop+gngm/original	0.5050	0.4740	**0.2999**
qrefs/combined	0.5075	0.4665	0.2986
median	**0.5450**	0.4559	0.2806

The results from Table 5.3 show that the top query reformulations, identified in the in-depth analysis from Section 5.2, remain effective for precision-oriented measures in clinical trials retrieval. Indeed, all the knowledge-enhanced models outperform the baseline by a margin greater than 2% for P@10. On the other hand, the improvements are less marked for infNDCG and Rprec – where only the model employing “solid” expansion (solid/original) and the combined model (qrefs/combined) outperform the baseline. We attribute the drop in performance of the other knowledge-enhanced models to the use of reduction techniques. In fact, reducing queries helps to focus more on relevant terms – thus increasing precision – but can hamper recall, as fewer terms are used to perform retrieval. Thus, the results suggest that the developed query reformulations are precision-oriented rather than recall-oriented.

To better understand the performances of the considered query reformulations on the Clinical Trials task, we perform a per-topic analysis that compares, for each measure, the five models with the task median values. Figures 5.4–5.8 display, topic by topic, the difference in performance between each model and the median values. For a positive difference (model better than median), a green barplot is shown, whereas for a negative difference (model worse than median), a red barplot is shown.

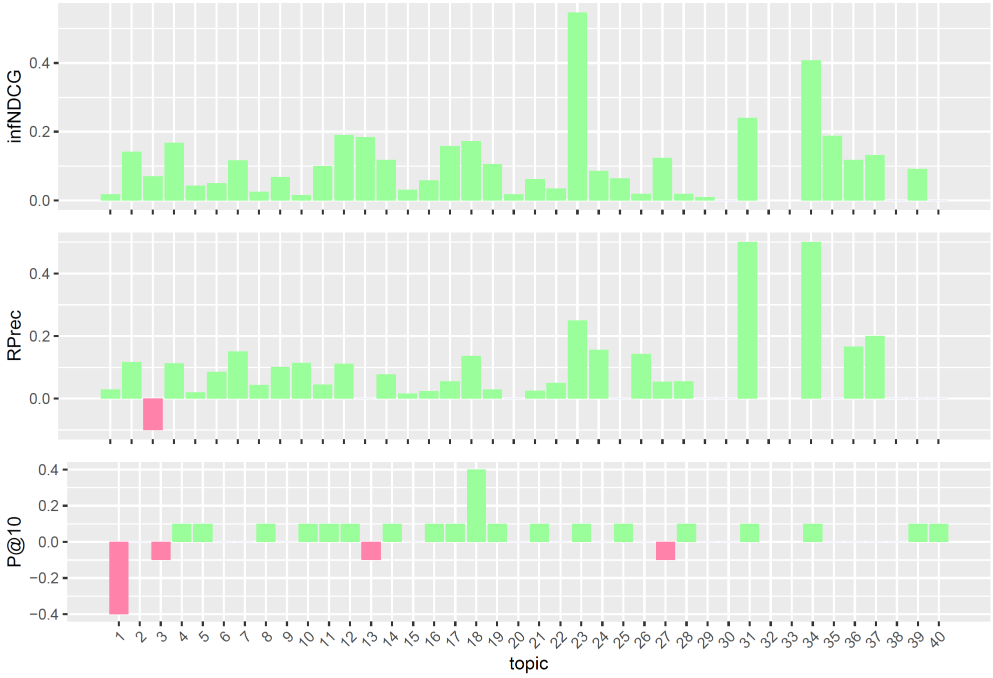


Fig. 5.4 Per-topic difference between base model and clinical trials median values. For a positive difference (model better than median), a green barplot is shown, whereas for a negative difference (model worse than median), a red barplot is shown.

The analysis of Figures 5.4–5.8 highlights an interesting scenario. First, all the considered models achieve performances higher than or equal to median values for most topics. Secondly, different knowledge-enhanced models achieve top performances on different topics. In other words, it does not exist a query reformulation that provides consistently better results than all the other ones. This is an interesting outcome, as it shows that the use

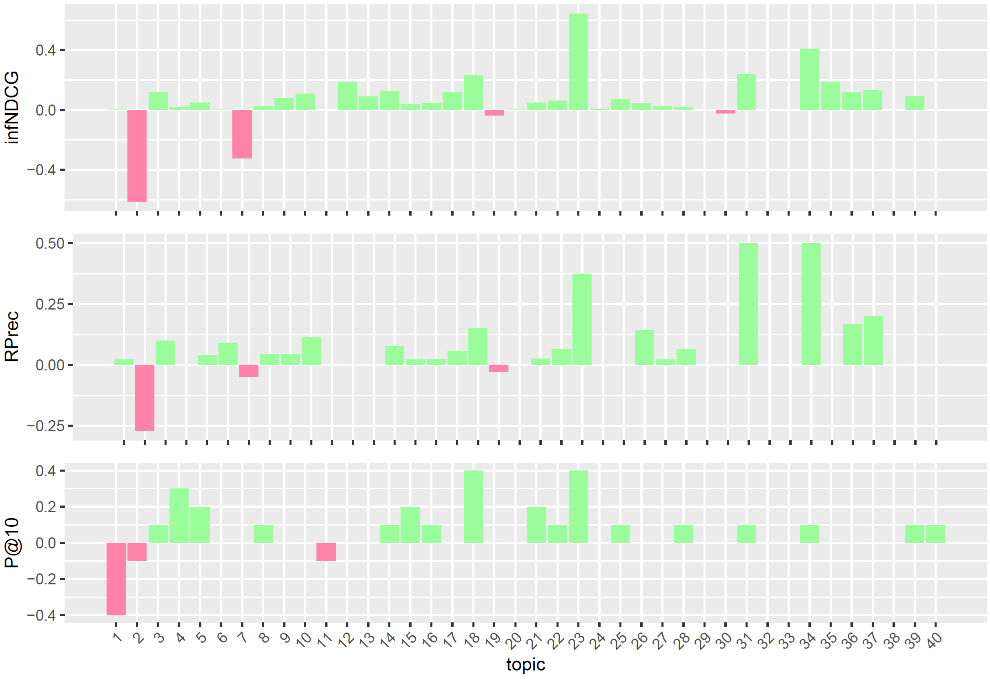


Fig. 5.5 Per-topic difference between neop/reduced model and clinical trials median values. For a positive difference (model better than median), a green barplot is shown, whereas for a negative difference (model worse than median), a red barplot is shown.

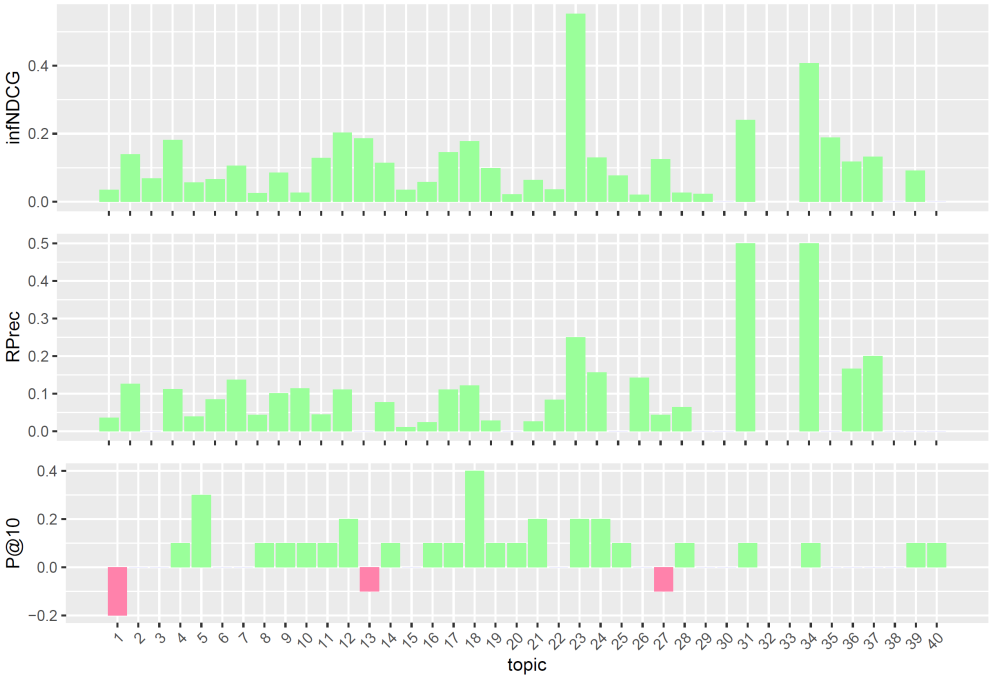


Fig. 5.6 Per-topic difference between solid/original model and clinical trials median values. For a positive difference (model better than median), a green barplot is shown, whereas for a negative difference (model worse than median), a red barplot is shown.

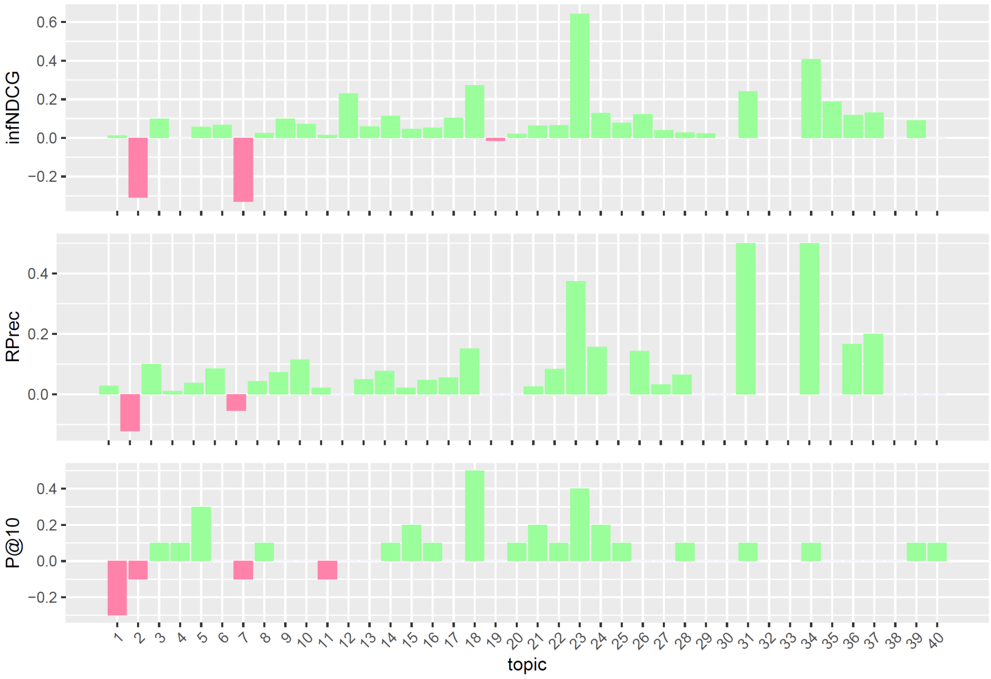


Fig. 5.7 Per-topic difference between solid/reduced model and clinical trials median values. For a positive difference (model better than median), a green barplot is shown, whereas for a negative difference (model worse than median), a red barplot is shown.

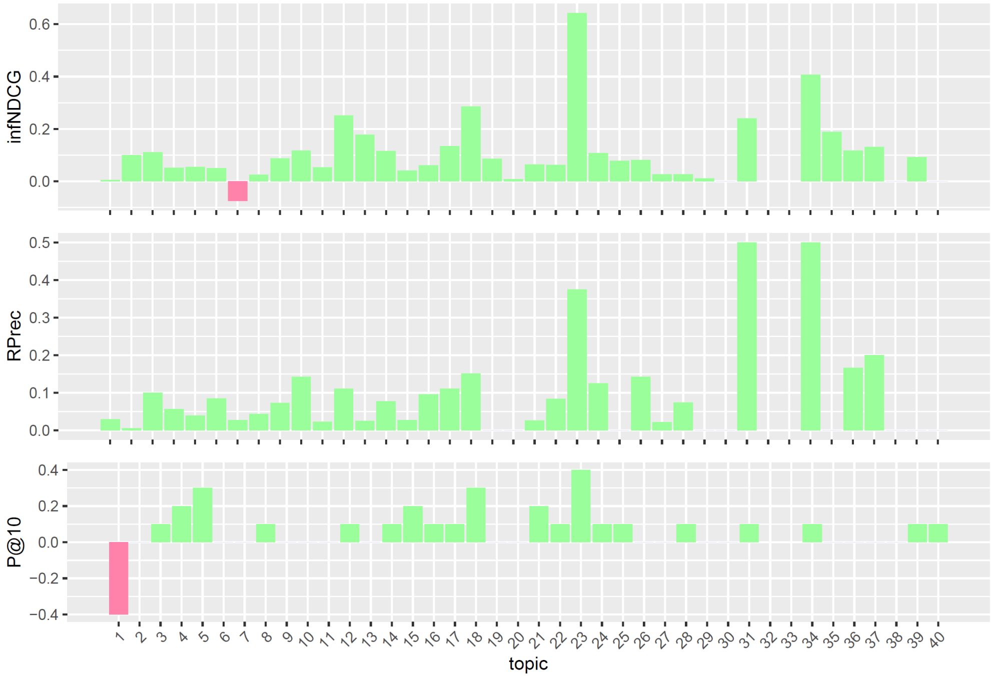


Fig. 5.8 Per-topic difference between qrefs/combined model and clinical trials median values. For a positive difference (model better than median), a green barplot is shown, whereas for a negative difference (model worse than median), a red barplot is shown.

of different knowledge-based query reformulations improves performances from different angles. However, the results obtained using CombSUM (i.e., qrefs/combined) suggest that more advanced techniques are required to effectively combine the different signals provided by the considered query reformulations. Although effective, CombSUM does not outperform all the individual models.

Regarding scientific literature retrieval, the results from Table 5.4 highlight a lower impact of the considered query reformulations on retrieval performances. In particular, none of the considered knowledge-enhanced models outperform the baseline for infNDCG. A possible reason for the marginal impact of query reformulations could lie on the shift from Whoosh to ElasticSearch for the indexing, retrieval, and filtering steps. Indeed, Whoosh and ElasticSearch handle the various steps required by our procedure differently, in particular query term weighting operations. However, further analyses are required to confirm this intuition – which are out of scope given the objectives of this thesis.

Comparison with TREC PM 2019 Top Systems

When looking at the detailed analysis in the TREC PM 2019 overview [187], we observe that the best performances obtained by our models surpass the top 10 threshold in both tasks for all the evaluation measures but one. For the Clinical Trials task, the model that relies on "solid" expansion (i.e., solid/original) achieves the second best performance for infNDCG and Rprec, and the third best for P@10 (see "BM25solid01o" in Table 6 Clinical Trials of the TREC PM 2019 overview [187]). As for Scientific Literature, the baseline model and the model empolying *neop* and *gngm* expansions achieve top 10 performances for infNDCG and Rprec, respectively (see "BM25" and "BM25neopgngm" in Table 6 Literature Articles of the TREC PM 2019 overview [187]). On the other hand, all the proposed models achieve performances lower than median values for P@10 – which is consistent with the results found in Section 5.2 for the 2018 Scientific Literature task.

Thus, the outcomes of this study highlight the effectiveness of the proposed query reformulations for retrieving relevant clinical trials in top positions of the ranking list. This is a promising result for at least two reasons. The first reason is that the proposed query reformulations, along with the weighting scheme applied to expansion terms, prove to be consistent across the years. The second reason regards the robustness of our approach. Indeed, the variation of the performance across topics for solid/original is smaller than any other top 10 system of TREC PM 2019 (see "BM25solid01o" in Figure 3 of the TREC PM 2019 overview [187]). Therefore, the developed query reformulations can be used to build knowledge-enhanced models that are robust to topic variations.

In the next section, given the consistent results achieved by the proposed query reformulations for the Clinical Trials task across the three years of TREC PM, we perform an a posteriori analysis focusing on clinical trials retrieval. The analysis provides an overview of the effectiveness of such techniques over the three years of TREC PM and aims to identify a robust subset of query reformulations specifically tailored to clinical trials retrieval.

5.4 A Posteriori Analysis of Query Reformulations

Based on the results achieved for the Clinical Trials task in Sections 5.2 and 5.3, we perform an a posteriori analysis on the effectiveness of the proposed query reformulations for clinical trials retrieval over the three years of TREC PM. This systematic analysis compares our approach and those proposed by the research groups that participated in all the three years of TREC PM.

5.4.1 Methodology

We adopt the methodology proposed in Section 5.2 for the Clinical Trials task, that is: indexing, query reformulation, retrieval, and filtering.

5.4.2 Experimental Setup

Test Collections and Knowledge Resources

We consider TREC PM 2017 (PM17) [186], 2018 (PM18) [185], and 2019 (PM19) [187] Tracks. We perform experiments on Clinical Trials collections using the 30, 50, and 40 topics provided, respectively, in 2017, 2018, and 2019. For query expansion, we adopt the following knowledge resources: NCI thesaurus [205], MeSH thesaurus [146], SNOMED CT [66], and UMLS metathesaurus [29]. For each resource, we consider the version contained within the 2018AA release of UMLS. For query reduction, we rely on the Cancer Biomarkers database [219]. We adopt the database version of January 17, 2018.

Evaluation Measures

We use the official measures adopted in the TREC PM Tracks, which are infNDCG, Rprec, and P@10. We do not report P@5 and P@15 since they were used only for the 2017 task, and we do not compute infNDCG for the 2017 task because the sampled relevance judgments are not available.

Experimental Procedure

For the 2017 and 2018 tasks, we rely on the top 5 query reformulations, ordered by P@10, found in the in-depth analysis performed in Section 5.2. Then, we apply the top 3 reformulations found in the 2017 and 2018 tasks to the 2019 task. We adopt Whoosh to perform indexing, retrieval, and filtering, and we set the rest of the parameters as in Sections 5.2 and 5.3.

5.4.3 Experimental Results

In Table 5.5, we report the results of our experiments on query reformulation (Part A) and compare them with the results obtained by the research groups that participated at TREC PM 2017, 2018 and 2019 (Part B). For 2017 and 2018 tasks, we present the five query reformulations with the highest P@10. Then, we report the effectiveness of the considered reformulations for the 2019 task. Each line shows a particular combination (*y*es or *n*o values) of semantic types (*neop*, *comd*, *gngm*), usage and expansion of `<other>` field (oth, oth_exp), query reduction (orig), and expansion using the weighted "solid" (tumor) keyword. We use the symbol '·' to indicate that the features oth, oth_exp are not applicable for the years 2018 and 2019 due to the absence of the `<other>` field in 2018 and 2019 topics. We highlight in **bold** the top 3 scores for each measure, and we use the symbol ‡ to indicate a combination that performs well in all three years. For the TREC PM systems, we select systems from those participants who submitted runs in all three years and reached top 10 performances in at least one edition for each measure [186, 185, 187]. The results reported in part B of Table 5.5 indicate the best score obtained by a particular system for a specific measure – again, note that the best results of a participant's system are often related to different runs. The symbol '—' means that the measure is not available, while '<' indicates that none of the runs submitted by the participant achieved top 10 performances. For the sake of comparison, we add for each measure the lowest score required to enter the top 10 TREC results list, and the score obtained by our best combination. The combination is indicated by the line number, which refers to its position in Part A of Table 5.5.

Analysis of Query Reformulations

The results from Table 5.5 (Part A) highlight that the use of "solid" expansions, as well as query gene reductions (orig = *n*), seems to consistently improve performances in 2017 – two of the three best combinations in terms of P@10 (lines 1 and 2) apply both techniques. Regarding knowledge-based expansions, the semantic type *gngm* (lines 1 and 5) seems more effective than *neop* (line 3), whereas *comd* does not seem to have any positive effect at all.

Table 5.5 Results for the TREC PM Clinical Trials tasks. Part A (top) reports the results achieved using the five most effective query reformulations for each year. (‡) indicates a particular query reformulation effective in all three years. Part B (bottom) reports the results obtained by participants' systems, along with the lowest score required to enter the top 10 TREC results list and the score obtained by the best combination of our approach. Further details are reported in Subsection 5.4.3.

A: Analysis of Query Reformulations

line	year	neop	comd	gngm	oth	oth_exp	orig	solid	P@10	infNDCG	Rprec
1	2017	n	n	y	n	n	n	0.1	**0.3931**	-	**0.3263**
2‡	2017	n	n	n	n	n	n	0.1	**0.4034**	-	**0.3361**
3	2017	y	n	n	n	n	n	0.1	0.3862	-	0.3202
4	2017	n	n	n	n	n	n	n	**0.3931**	-	0.3241
5	2017	n	n	y	n	n	y	n	0.3862	-	**0.3243**
6	2018	n	n	n	·	·	y	n	0.5680	**0.5411**	**0.4197**
7	2018	n	n	n	·	·	y	0.1	**0.5740**	**0.5403**	**0.4179**
8	2018	y	n	n	·	·	n	n	**0.5700**	0.5345	0.4134
9‡	2018	n	n	n	·	·	n	0.1	**0.5820**	**0.5446**	**0.4205**
10	2018	y	n	y	·	·	n	n	0.5680	0.5393	0.4122
11	2019	n	n	n	·	·	y	0.1	**0.5368**	**0.6239**	**0.4386**
12	2019	y	n	n	·	·	n	n	0.5237	0.5755	0.4135
13‡	2019	n	n	n	·	·	n	0.1	**0.5316**	**0.5940**	**0.4264**
14	2019	n	n	y	·	·	n	0.1	**0.5263**	**0.6070**	**0.4302**
15	2019	n	n	n	·	·	n	n	0.5105	0.5853	0.4239

B: Comparison with TREC PM other Participants

line	year	TREC PM Participant Identifier	P@10	infNDCG	Rprec
1	2017	BiTeM	0.3586	-	-
2	2017	cbnu	<	-	-
3	2017	CSIROmed	<	-	-
4	2017	ECNUica	<	-	-
5	2017	Poznan	0.3690	-	-
	2017	Top 10 threshold	0.3586	-	-
	2017	Best combination of our approach	(A.2‡) 0.4034	-	0.3361
6	2018	BiTeM	<	<	<
7	2018	cbnu	<	<	<
8	2018	CSIROmed	<	<	<
9	2018	ECNUica	<	<	<
10	2018	Poznan	0.5580	0.4894	0.4101
	2018	Top 10 threshold	0.5240	0.4736	0.3658
	2018	Best combination of our approach	(A.9‡) 0.5820	0.5446	0.4205
11	2019	BiTeM	0.4711	0.4963	0.3698
12	2019	cbnu	0.4921	0.5568	0.4121
13	2019	CSIROmed	0.4921	0.4930	0.3586
14	2019	ECNUica	0.5053	0.5355	0.4001
15	2019	Poznan	0.4421	0.4810	0.3503
	2019	Top 10 threshold	0.3658	0.4320	0.3230
	2019	Best combination of our approach	(A.11) 0.5368	0.6239	0.4386

All five combinations do not consider the `<other>` field (oth = *n*) nor its expansion (oth_exp = *n*) – confirming our intuition that it might represent a potential source of noise in retrieving precise information for patients. Similarly to 2017, two of the three best combinations in 2018 do not use knowledge-based expansions and rely on "solid" (tumor) expansion (lines 7 and 9). In particular, the reformulation combining query gene reductions and "solid" expansion (marked as ‡) provides the best performances for all the measures considered, both in 2017 and 2018. This suggests that removing over-specialized information (i.e., the gene mutations) or adding general terms (e.g., solid) benefits the retrieval. A possible reason is related to the nature of the document sets, since clinical trials often contain general requirements to allow patients to enroll. The results obtained in 2019 with the top 3 query reformulations from both 2017 and 2018 confirm this trend. The reformulation combining query gene reductions and "solid" expansion (line 13‡) achieves top 3 performances in 2019, however two query reformulations from 2017 (line 14) and 2018 (line 11) provide better performances. This result shows how difficult the task is. Indeed, even though we found a particular query reformulation approach (marked as ‡) to be highly effective in all three years – especially in 2017 and 2018 – it was not the best approach for 2019.

Therefore, this analysis helps to identify a robust subset of query reformulations for clinical trials retrieval. The selected query reformulations can be used at the early stages of the IR pipeline to retrieve relevant clinical trials in top positions of the ranking list. In this way, the different signals, that (knowledge-based) query reformulations provide, can be used (and combined) by multi-stage IR systems to obtain a richer pool of relevant documents, thus reducing the semantic gap between queries and documents.

Comparison with TREC PM Systems

The results from Table 5.5 (Part B) mark a clear division between the 2017 and 2018 tasks and the 2019 task. In 2017 and 2018, most of the participants' runs do not reach the top 10 threshold in any of the considered measures – the only exception is the research group from Poznan University of Technology, whose best runs always belong to the top 10 performing runs for the task. Conversely, in 2019 all the participants' best runs achieve results higher than the top 10 threshold. The reason behind the improvement of participants' runs in 2019 mostly relates to the use of supervised re-ranking models, which exploit relevance judgments from previous years for training. Thus, participants' approaches consist of expensive supervised multi-stage systems that, unlike ours, require relevance labels to work.

When we consider the results obtained using the query reformulations from Table 5.5 (Part A), we see that all query reformulations obtain results higher than the top 10 threshold for all the considered measures in all three years. Furthermore, query reformulations consistently achieve better results than participants' systems for each measure in all three years. Besides, unlike participants's systems, our approach operates only in the early stages of the IR pipeline and does not require any labeled data to work. This is an indication of the robustness of our approach across the different collections and also of the effectiveness of the proposed query reformulations for clinical trials retrieval. In particular, it is worth mentioning that models using the (‡) query reformulation achieve performances that belong to the top 3 of the best-performing systems in each year of the TREC PM Track [186, 185, 187].

5.5 Chapter Outcomes and Lessons Learned

In this chapter, we investigated how to use external knowledge resources to enhance bag-of-words representations and reduce the effect of the semantic gap between queries and documents, focusing on an important use-case in medical IR: providing useful precision medicine information to clinicians treating cancer patients. To this end, we developed knowledge-enhanced lexical models that integrate external knowledge in the retrieval stage through query reformulation techniques.

As a first step, we have conducted a preliminary study on the TREC Precision Medicine (PM) 2018 Clinical Trials task – where the objective is to retrieve relevant clinical trials for which the patient is eligible. We proposed a procedure to: 1) expand queries iteratively, relying on medical knowledge resources [122, 155], to increase the probability of finding relevant trials by adding neoplastic, genetic, and proteic term variants to the original query; and 2) filter out trials, based on demographic data, for which the patient is not eligible. The experimental results showed that retrieval models perform best when none of the developed query expansions are used. The reasons behind the detrimental effect of the proposed query expansions are two: (i) the lack of an appropriate weighting scheme on query terms, (ii) the use of all the knowledge resources contained within UMLS – regardless of their relevance to the considered task.

Thus, we have deepened the analysis performed in the preliminary study and we have extended it to both scientific literature and clinical trials retrieval. In other words, we took advantage of the dual nature of TREC PM collections and we evaluated several state-of-the-art query expansion and reduction techniques to examine whether a particular approach can be helpful for both scientific literature and clinical trials. The analysis showed that no clear pattern emerges for both tasks. Overall, a query expansion approach using a selected set of semantic types helps the retrieval of scientific literature. On the other hand, a query reduction approach and a "solid" (tumor) expansion improve performances on clinical trials retrieval. Nevertheless, most of the proposed query reformulations perform well for both tasks. Besides, we found that a particular combination (marked as ‡ in Table 5.2) could have been one of the top 10 performing runs for many evaluation measures in both TREC PM 2017 and 2018. Hence, the in-depth analysis, that stemmed from our preliminary study, highlighted the effectiveness of applying a weighting scheme on expansion terms and selecting tailored knowledge resources for query expansion and reduction techniques.

Given the outcomes of the in-depth analysis, we have conducted a validation study on the TREC PM 2019 Track. We performed experiments on both tasks, with a particular focus on the Clinical Trials task, to evaluate how the different query reformulations – tested on previous TREC PM collections – affect the results and whether the findings obtained in the previous analysis remain valid. The experimental results highlighted the effectiveness of the tested query reformulations for retrieving relevant clinical trials – especially in top positions of the ranking list. This proves that, across the years, the tested query reformulations remain effective. Furthermore, the per-topic analysis showed that different knowledge-enhanced models achieve top performances on different topics. In other words, it does not exist a query reformulation that consistently provides better results than all the other ones. Therefore, the combination of several knowledge-based query reformulations – which focus on different aspects of the queries and promote complementary information – can improve performances from different angles.

Finally, we have performed an a posteriori analysis on the effectiveness of the proposed query reformulations for clinical trials retrieval. We compared our approach and those proposed by the research groups that participated in all the three years of TREC PM. The experimental results confirmed the effectiveness of the proposed query reformulations in all collections. Besides, the analysis helped to identify a robust subset of query reformulations for clinical trials retrieval. The selected query reformulations can be used at the early stages of the IR pipeline to retrieve relevant clinical trials in top positions of the ranking list. In this way, the different signals that (knowledge-based) query reformulations provide can be used (and combined) by multi-stage IR systems to obtain a richer pool of relevant documents, thus reducing the semantic gap between queries and documents.

Chapter 6

Knowledge-Enhanced Semantic Models

Since the advent of word2vec [163, 162] and doc2vec [138] models, distributed representations of words and documents have experienced widespread use in NLP and IR tasks. Nevertheless, even though word2vec and doc2vec models effectively encode semantic and syntactic relationships relying on the distributional hypothesis [102], they fail to capture relational information – e.g., synonymic dependencies – for words not occurring in the same context [119]. In this regard, many efforts have been made by the NLP community to integrate the relational information – contained within external knowledge resources – in the learning process of word and document representations, such as [56, 250, 74, 156, 204], to name a few. On the other hand, as we have shown in Subsection 3.5.2, fewer studies have been conducted in IR to investigate how relational information can be incorporated within the word and document representations generated by neural language models [147, 169, 170, 220].

Besides, even though knowledge-enhanced neural language models have been proven effective in many NLP tasks, their effectiveness is limited in IR – as we will see throughout this chapter. In particular, we identify two reasons causing this performance gap. First, knowledge-enhanced neural language models have been used in IR mostly for re-ranking [147, 170]. In re-ranking, knowledge-enhanced neural language models are limited to candidate documents retrieved by lexical (bag-of-words) models, which are not suited to address the semantic gap. Thus, relevant documents most affected by the semantic gap – that knowledge-enhanced models could help identify – simply remain undiscovered. Secondly, IR tasks are different from NLP tasks. IR requires to match a given query to a set of relevant documents, whereas NLP mostly deals with the discovery of semantic and linguistic regularities. Therefore, (knowledge-enhanced) neural language models do not encode relevance signals or discriminative aspects between queries and documents, which are fundamental to effectively address IR tasks.

To investigate and address the above limitations, we focus on medical literature and we consider two linguistic features related to the semantic gap: synonymy and polysemy. Within medical literature, the large presence of synonymous and polysemous words – along with the use of acronyms and morphosyntactic variants – poses a critical challenge to retrieval models. For example, an IR model might not effectively answer a query related to the concept of "tumor" if it does not identify the synonymy relationship occurring between "tumor" and, say, "neoplasm". On the other hand, an IR model might retrieve erroneous documents for a query related to the concept of "common cold" if it does not distinguish between the different contextual meanings of the word "cold". In fact, "cold" assumes different meanings depending on the context, including: common cold, cold temperature, and COLD as per Chronic Obstructive Lung Disease. We refer to these queries as semantically hard queries.

Thus, to fully understand knowledge-enhanced neural language models and their effectiveness for IR tasks, we perform a reproducibility study of the works by Liu et al. [147] and Nguyen et al. [169]. The knowledge-enhanced word embeddings used, and proposed, by Liu et al. [147] extend word2vec models [162] by applying a knowledge-based regularization during/after the training of word representations. The assumption is that related words in an external knowledge resource – e.g., synonyms – should have similar representations in the latent space. The learned representations encode synonymy and are then used to perform re-ranking on medical IR collections. On the other hand, the work by Nguyen et al. [169] attempts to integrate knowledge-based information in the learning process of document representations. In particular, the authors investigate how to optimize document representations learned by doc2vec models [138], relying on concepts – specified within external knowledge resources – and words – belonging to the target corpus. The learned representations encode both synonymy and polysemy and are then used to perform query expansion, with the objective of improving the retrieval performance of lexical models on medical IR collections.

Then, motivated by the outcomes of the reproducibility study, we pose the following research questions:

RQ1 Which feature between synonymy and polysemy can be exploited to reduce the semantic gap and improve retrieval?

RQ2 How can external knowledge resources help to bridge the semantic gap between queries and documents?

For **RQ1**, we investigate how to leverage synonymy and polysemy in the learning process of neural models. How can we model both features jointly? Which feature is prominent for retrieval effectiveness? To what extent modeling these features is effective for semantically hard queries?

For **RQ2**, we explore how integrating external knowledge into neural models impacts retrieval performances. In particular, we seek to understand whether knowledge-enhanced neural models retrieve relevant documents that are most affected by the semantic gap. In other words, to what extent knowledge-enhanced neural models retrieve different relevant documents?

To address the research questions, we propose the Semantic-Aware Neural Framework for IR (SAFIR), an unsupervised knowledge-enhanced neural framework for IR. SAFIR jointly learns word, concept, and document representations from scratch. The learned representations are optimized for IR and encode both polysemy and synonymy to address the semantic gap between queries and documents. SAFIR can be applied to any domain where external knowledge resources are available (e.g., medical, legal, news), and it does not require any labeled data for training – which are scarce and expensive resources.

We conduct an experimental evaluation to compare SAFIR with knowledge-enhanced neural language models on medical literature retrieval, a specific task of Clinical Decision Support (CDS). We adopt the UMLS metathesaurus [29] as external knowledge resource and we evaluate models on the TREC CDS collections [188, 189, 184] and on the OHSUMED collection [107].

We consider two retrieval strategies to investigate the research questions: document retrieval and query expansion. Document retrieval gives us the opportunity to investigate the effectiveness of integrating external knowledge into neural models for the typical retrieval scenario, where systems retrieve a set of candidate documents given a query. Query expansion allows us to investigate the effectiveness of knowledge-enhanced neural models – which are specifically designed to address the semantic gap – in retrieving feedback documents for PRF based methods. In other words, we evaluate if knowledge-enhanced neural models provide expansion terms that are more effective at reducing the semantic gap for lexical models.

The main contributions of this chapter are:

- **C1** We perform a reproduciblity study of two seminal works on knowledge-enhanced neural models for IR. The outcomes of the reproducibility study highlight the potential of knowledge-enhanced models – but also the limitations of neural language models for IR.

- **C2** We present SAFIR, an unsupervised knowledge-enhanced neural framework for IR. To the best of our knowledge, SAFIR is the first unsupervised framework that models synonymy and polysemy to jointly learn word, concept, and document representations specifically for IR. SAFIR does not require any labeled data for training and can be used in domains where explicit relevance labels are scarce and expensive resources.

C3 We show how SAFIR integrates synonymy and polysemy for IR tasks. Furthermore, we perform extensive quantitative and qualitative analyses which provide insights into the individual and joint impact of these features in IR. In particular, we investigate the effectiveness of modeling synonymy and polysemy to answer semantically hard queries.

C4 We perform quantitative and qualitative analyses that investigate the ability of knowledge-enhanced neural models to retrieve relevant documents affected by the semantic gap. Furthermore, we evaluate the degree of similarity between SAFIR and the considered baselines to understand to what extent they retrieve different relevant documents.

C5 We perform in-depth analyses to evaluate the effectiveness of SAFIR compared to the considered baselines and show its robustness for most collections. The analysis for query expansion highlights that knowledge-enhanced neural models grasp different signals than lexical models and retrieve feedback documents that are more effective in providing expansion terms for PRF based methods.

The rest of this chapter is as follows. We introduce the required notation in Section 6.1. We perform the reproducibility study of the works by Liu et al. [147] and Nguyen et al. [169] in Sections 6.2 and 6.3, respectively. We discuss the outcomes of the reproducibility study in Section 6.4, where we also analyze the limitations of the considered approaches. Then, in Section 6.5, we present SAFIR, our novel unsupervised knowledge-enhanced neural framework. In Section 6.6, we describe the experimental setup. We report the experimental results and provide in-depth quantitative and qualitative analyses for document retrieval and query expansion in Sections 6.7 and 6.8, respectively. Finally, in Section 6.9, we conclude the chapter with a discussion on the lessons learned where we highlight the take-home messages.

6.1 Notation

We call D the set of corpus documents and V the set of unique words in the vocabulary. A document is a sequence of words $d = (w_j)_{j=1}^{m}$, where w_j is the word in the j^{th} position of d and $m = |d|$ is the document length. Similarly, a query is a sequence of words $q = (w_i)_{i=1}^{n}$, where w_i is the word in the i^{th} position of q and $n = |q|$ is the query length.

A knowledge resource is a graph $\Omega = (\mathscr{C}, \mathscr{E})$, where $\mathscr{C}$ is the set of nodes (i.e., concepts) and $\mathscr{E}$ is the set of edges (i.e., relations between concepts). Given Ω, we derive the meaning of a word w in d by associating w to a concept $c \in \mathscr{C}$ based on the context of w. Therefore, we do not consider phrase-concept associations and we refer to words or terms interchangeably.

We define a knowledge-enhanced document $\phi = (\langle w_j, c_j \rangle)_{j=1}^{m} \in \Phi$ to be an ordered sequence of contextualized word-concept pairs where $w_j \in V$, $c_j \in \mathscr{C}$, and Φ is the set of knowledge-enhanced documents. Symmetrically, a knowledge-enhanced query is defined as $\varphi = (\langle w_i, c_i \rangle)_{i=1}^{n}$.

Given $\Omega = (\mathscr{C}, \mathscr{E})$, we define $C \subseteq \mathscr{C}$ as the set of unique concepts associated with the words contained in the corpus and we consider as synonyms all the semantic and terminological variants that express a concept $c \in C$. This means that also acronyms, graphical variants, and morphosyntactic variants are considered synonyms.

6.2 Knowledge-Enhanced Word Embeddings for IR

We describe the knowledge-enhanced word embeddings used, and proposed, by Liu et al. [147]. The considered approaches extend word2vec models [162] by applying a knowledge-based regularization during/after the training of word representations. The problem can be defined as a multi-task learning problem, where distributional-based and relational-based objectives are modeled together using the same set of shared word embeddings. Therefore, we can divide the approaches considered by Liu et al. [147] into three main categories, based on the different learning techniques: alternate learning, joint learning, and retrofitting.

Since the considered word2vec architecture is CBOW, we briefly recall its main aspects and then present the proposed extensions. Given a word vocabulary $V = \{w_1, ..., w_n\}$ and a document collection defined as a sequence of words $w_1, ..., w_T$, word2vec CBOW learns word representations so as to maximize the log-likelihood of each word w_t (the t^{th} word within the sequence) given its context, i.e., the set of words within a window of size k centered at w_t (w_t excluded):

$$\frac{1}{T} \sum_{t=1}^{T} \log p(w_t | w_{t-k}^{t+k}) \tag{6.1}$$

The probability of a word given its context is defined as:

$$p(w_t | w_{t-k}^{t+k}) = \frac{\exp(\boldsymbol{w}_t^{\top} \boldsymbol{w}_{t-k}^{t+k})}{\sum_{i \in V} \exp(\boldsymbol{w}_i^{\top} \boldsymbol{w}_{t-k}^{t+k})}, \quad \boldsymbol{w}_{t-k}^{t+k} = \sum_{j=t-k, j \neq t}^{t+k} \boldsymbol{w}_j \tag{6.2}$$

where the context representation $\boldsymbol{w}_{t-k}^{t+k}$ is the sum of the word representations for the words occurring in the context window.

6.2.1 Alternate Learning

Proposed by Yu and Dredze [250], the model combines word2vec with a Relation Constrained Model (RCM) that, given an external knowledge resource Ω encoding semantic relations, learns word representations by predicting one word from a related word within Ω. Thus, the joint model maximizes the following linear combination of word2vec and RCM objectives:

$$\frac{1}{T}\sum_{t=1}^{T}\log p(w_t|w_{t-k}^{t+k}) + \frac{\lambda}{|R|}\sum_{(w_i,w_j)\in R}\log p(w_i|w_j) \tag{6.3}$$

where $R = \{(w_i,w_j)\in V\times V|\ w_i,w_j\in c \vee (w_i\in c_i \wedge w_j\in c_j \wedge (c_i,c_j)\in\mathscr{E})\}$ and λ is a hyperparameter controlling the strength of RCM.[1] The model is trained in an alternate learning fashion, where the RCM objective is computed every m words. Word representations are learned using stochastic gradient ascent with two learning rates, η_{CBOW} and η_{RCM}, which are updated separately in turn. As stated by Liu et al. [147], the risk of this process is that the RCM update could undo the word2vec update, as the distributional context of a word is no longer taken into account when regularizing over Ω relations.

6.2.2 Joint Learning

Proposed by Liu et al. [147], the model combines the word2vec objective with the requirement that if a word can be well generated from a given context, its related words in Ω should also be well generated from the same context:

$$\frac{1}{T}\sum_{t=1}^{T}\left[\log p(w_t|w_{t-k}^{t+k}) - \alpha\sum_{w_s:(w_t,w_s)\in R}\rho(w_s|w_t)\left[\log p(w_t|w_{t-k}^{t+k}) - \log p(w_s|w_{t-k}^{t+k})\right]^2\right] \tag{6.4}$$

where α is a weighting hyperparameter and $\rho(w_s|w_t) = \frac{ttf(w_s)}{\sum_{w_j:(w_t,w_j)\in R} ttf(w_j)}$ is the relative frequency of w_s in the document collection based on the total term frequency $ttf(\cdot)$. The weighting scheme $\rho(\cdot|\cdot)$ diversifies the importance of relations between words depending on their frequency within the collection. The model is trained in a joint learning fashion, where the objectives are computed together and all the gradients and updates are done with respect to both functions at the same time.

[1] By a slight abuse of notation, we use $w\in c$ to indicate that the word w expresses the concept c.

6.2.3 Retrofitting

Proposed by Faruqui et al. [74], the model learns word representations that are both close (under a distance metric) to the data-driven representations obtained by word2vec and to the representations of adjacent vertices in Ω. Thus, given the matrix $\hat{\boldsymbol{U}}$ of vector representations $\hat{\boldsymbol{u}}_i \in \mathbb{R}^a$, for each $w_i \in V$, learned using a standard data-driven architecture (e.g., CBOW), where a denotes the size of the word vectors, the model learns the matrix $\boldsymbol{U} = (\boldsymbol{u}_1, ..., \boldsymbol{u}_n)$ by minimizing the following loss function:

$$\Psi(\boldsymbol{U}) = \sum_{i=1}^{n} \left[\omega_i ||\boldsymbol{u}_i - \hat{\boldsymbol{u}}_i||^2 + \sum_{w_j:(w_i,w_j)\in R} \beta_{ij} ||\boldsymbol{u}_i - \boldsymbol{u}_j||^2 \right] \tag{6.5}$$

where ω_i and β_{ij} values control the relative strengths of associations and the distance metric is defined to be the Euclidean distance. The model first learns word representations independent of the information in Ω and then retrofits them. The retrofitting optimization is performed through an efficient iterative updating method [25, 216, 54, 55], which eventually leads to the following update:

$$\boldsymbol{u}_i = \frac{\sum_{w_j:(w_i,w_j)\in R} \beta_{ij} \boldsymbol{u}_j + \omega_i \hat{\boldsymbol{u}}_i}{\sum_{w_j:(w_i,w_j)\in R} \beta_{ij} + \omega_i} \tag{6.6}$$

A modified version of this approach has been proposed by Liu et al. [147], where they apply the weighting scheme $\rho(\cdot|\cdot)$ to the relational-based component of the loss function $\Psi(\boldsymbol{U})$:

$$\Psi(\boldsymbol{U}) = \sum_{i=1}^{n} \left[||\boldsymbol{u}_i - \hat{\boldsymbol{u}}_i||^2 + \beta \sum_{w_j:(w_i,w_j)\in R} \rho(w_j|w_i) ||\boldsymbol{u}_i - \boldsymbol{u}_j||^2 \right] \tag{6.7}$$

6.2.4 Experimental Setup and Implementation Details

Experimental Setup

We reproduce the experiments performed by Liu et al. [147] on the OHSUMED collection [107]. We chose to reproduce the experiments on OHSUMED because it is a standard collection for medical literature retrieval, which has also been used in various related works [56, 169, 220]. In a personal communication with the authors, they confirmed that they relied on the 63 official topics and the *description* field to perform experiments. More details on the considered test collection can be found in Subsection 2.1.1.

Liu et al. adopt the UMLS metathesaurus [29] as Ω. Each UMLS concept, identified by a CUI, contains a set of term variants which are used as synonyms by Liu et al.[2] Since there

[2] Only single-word term variants are considered.

is no information about the UMLS version used by Liu et al., we rely on the UMLS 2018AA release. However, knowing the exact UMLS release is important to enable reproducibility, as the number of relations and concepts within UMLS is constantly changing. The reader can find a detailed description of UMLS in Subsection 2.2.4.

Word embeddings are used for re-ranking. Following the scheme proposed by Vulić and Moens [235], document and query representations are built by summing up all the word embeddings for words contained within the document and the query. For document representations, word embeddings are weighted by IDF. The obtained query and document representations are used to re-rank an initial pool of 1000 documents retrieved by a standard IR model as follows:

$$\text{score}(q,d) = \gamma \text{BoW}(q,d) + (1-\gamma)\text{sim}(q,d) \tag{6.8}$$

where γ is a combination hyperparameter, $\text{BoW}(\cdot,\cdot)$ is a lexical (bag-of-words) model, such as BM25 or QLM, and $\text{sim}(\cdot,\cdot)$ is a similarity function between query and document embeddings, such as cosine similarity. The scores obtained by $\text{BoW}(\cdot,\cdot)$ and $\text{sim}(\cdot,\cdot)$ are normalized using Min-Max normalization before being combined.

The evaluation measures used in the experiments are: P@10 and MAP. P@10 is used as the main performance indicator, whereas MAP is used as the second indicator because it has been adopted in the reference paper – even though the measure is known to have some limitations as highlighted by Fuhr [84]. Rather than using a two-tailed Student's t-test as in the reference paper, we perform a statistical significance analysis with Tukey's HSD test to assess the statistical significance of performance differences for the retrieval models. We recall that Tukey's HSD test is a viable method for dealing with the multiple comparisons problem [84, 37]. We apply the Tague-Sutcliffe transformation to Tukey's HSD test [218]. Further details on Tukey's HSD test are reported in Subsection 4.1.2.

Implementation Details

We implement word2vec and its knowledge-enhanced extensions from scratch in Python, relying on TensorFlow to build the network architectures. In the reference paper, there is no mention on how to map CUIs from UMLS to words within OHSUMED. Therefore, we rely on QuickUMLS [207], a fast unsupervised concept extractor built on UMLS (see Section 3.1), to map each word in V to a list of candidate CUIs. For each word, we keep the first occurrence within the candidates list as the word CUI and we adopt all its single-word term variants as synonyms.[3] We consider CUIs from all UMLS semantic types, not only from QuickUMLS default ones.

To train word2vec and its knowledge-enhanced extensions, we use Noise Constrastive Estimation (NCE) [98] rather than negative sampling [163]. We tested word2vec with both strategies, and we got the closest results to those reported by Liu et al. with NCE (see Table 6.2). As in the reference paper, we set the dimension of the embeddings $a = 300$, the context window size $k = 5$, and the number of negative samples equal to 10. The collection is not preprocessed before training, i.e. no stemming nor stopwords removal, but words appearing less than 5 times are removed. There is no mention to the library used to implement BM25 (with $k1 = 1.2$ and $b = 0.75$) and QLM (using Dirichlet smoothing with $\mu = 2000$), thus we rely on ElasticSearch.

The main challenge we found to reproduce the results obtained by Liu et al. lies in the choice of the parameters and hyperparameters for the knowledge-enhanced word embeddings. The reference paper lacks a comprehensive description of all the parameters and hyperparameters used by the different models – especially by those originally presented in [250, 74]. Therefore, to select the parameters and hyperparameters for this study, we relied both on the reference paper [147] and on [250, 74]. For each setting, we report the reference source too. We train word2vec, alternate, and joint learning models for a single epoch [250], while the retrofitted models are trained for 10 epochs [74]. We optimize all models using AdaGrad [67], as in [74]. Learning rates for word2vec and the joint learning model are set equal to 0.025 [147, 250], whereas the learning rate for the RCM model is set equal to 0.01 [250]. The λ hyperparameter controlling the strength of RCM in the alternate learning model is set to 1, and the RCM objective is computed every $m \approxeq 1,000$ words.[4] The ω_i and β_{ij} hyperparameters in the original retrofitting model [74] are set to 1 and to $\text{degree}(w_i)^{-1}$ (with w_i being the node the update is being applied to), respectively. Regarding the α and β hyperparameters of the joint learning and the modified retrofitting models proposed by Liu et al. [147], they have been optimized using 2-fold cross-validation and the results of this optimization are visible in Figures $1-4$ of the reference paper. Since the optimization is performed for each combination of the proposed models with the lexical models, we identify two sets of hyperparameters: $\{\alpha_{\text{BM25}} = 0.6, \beta_{\text{BM25}} = 0.6\}$, and $\{\alpha_{\text{QLM}} = 0.3, \beta_{\text{QLM}} = 0.5\}$. Conversely, we cannot adopt the same approach to identify the best values for the γ hyperparameter. However, the reference paper states that the best setting for γ is always around $0.5 - 0.6$. Therefore, for each combination of the word embedding models with the lexical models, we perform the re-ranking approach with $\gamma \in \{0.5, 0.55, 0.6\}$ and we select the value of

[3] Albeit being prone to ambiguous mappings, this efficient approach yielded effective results.

[4] We also tested the model using $m \approxeq 10,000$, as in [74], obtaining similar results.

γ that provides the best results in terms of P@10. The results of this optimization are shown in Table 6.1. Then, the reproduced models are compared with the original versions in Table 6.2, where for each re-ranking approach the combination hyperparameter γ is selected from Table 6.1. Finally, we also perform a comparison on the sensitivity of α and β hyperparameters for the original and reproduced versions of the joint learning and the modified retrofitting models. The behavior of the different versions can be found in Figure 6.2.

For consistency with the reference paper, we report the original names in all tables and figures. That is, QLM is called LM, word2vec is called CBOW, the alternate learning model is called Yu+BoW, the retrofitting model is called Faruqui+BoW, and the joint learning and modified retrofitting models are called Online+BoW and Offline+BoW, respectively.

6.2.5 Experimental Results

In Table 6.1, we present the results of the optimization for the re-ranking approach with $\gamma \in \{0.5, 0.55, 0.6\}$. For each model, we select the best γ configuration based on its performance in P@10. Ties are resolved looking at MAP scores. The best γ configurations are then used to perform a comparison with the results obtained by Liu et al. [147].

Table 6.1 Result comparison for the re-ranking method using different values of $\gamma \in \{0.5, 0.55, 0.6\}$. For each evaluation measure, the first column refers to $\gamma = 0.5$, the second column refers to $\gamma = 0.55$, and the third to $\gamma = 0.6$. For each re-ranking combination, **bold** values represent the best γ configuration based on P@10 and, when necessary, on MAP.

	P@10			MAP		
	$\gamma = 0.5$	$\gamma = 0.55$	$\gamma = 0.6$	$\gamma = 0.5$	$\gamma = 0.55$	$\gamma = 0.6$
CBOW+BM25	**0.5143**	0.5079	0.5079	0.3082	0.3105	0.3119
CBOW+LM	0.4444	0.4444	**0.4460**	0.2773	0.2805	0.2828
Yu+BM25	0.5143	0.5143	**0.5143**	0.3086	0.3109	**0.3112**
Online+BM25	**0.5206**	0.5190	0.5111	0.3078	0.3092	0.3091
Yu+LM	**0.4524**	0.4460	0.4429	0.2772	0.2806	0.2825
Online+LM	0.4524	**0.4524**	0.4508	0.2774	**0.2793**	0.2812
Faruqui+BM25	0.5143	0.5143	**0.5143**	0.3103	0.3117	**0.3134**
Offline+BM25	0.5127	0.5127	**0.5143**	0.3104	0.3126	0.3133
Faruqui+LM	**0.4540**	0.4492	0.4476	0.2797	0.2822	0.2835
Offline+LM	0.4460	**0.4476**	0.4429	0.2794	0.2823	0.2838

In Table 6.2, we present the results of the comparison between the original and the reproduced version of the models. First, we observe that we successfully reproduced word2vec (third line of Table 6.2). The absolute differences between the original and the reproduced versions of word2vec are below 0.02, for both P@10 and MAP. In particular, the difference in terms of P@10 is 0.0004. Secondly, we observe how the reproduced versions of BM25 and QLM provide higher results in both P@10 and MAP. The only absolute difference lower than 0.05 is for BM25 in MAP. This leads the subsequent re-ranking approaches to provide limited improvements over the lexical models compared to the original versions. Nevertheless, we can confirm the assumption made by Liu et al. [147] about the beneficial effect of constraining word embeddings by relational knowledge provided by an external knowledge resource. Indeed, both in the reference paper and here, the approach providing the best results in terms of P@10 is the combination of BM25 with the joint learning model. In our case, however, the BM25 combined with the retrofitting method by Faruqui et al. [74] provides better results in terms of MAP than the BM25 combined with the joint learning model.

Table 6.2 Result comparison between original version of the models (as in [147]) and their reproduced versions. For each evaluation measure, the first column reports the scores of the original model, the second column reports the scores of the reproduced version and the third column reports the difference between the original and reproduced versions; a negative difference indicates that the reproduced versions are stronger than those employed in the reference paper. For each re-ranking combination, the hyperparameter γ is selected from Table 6.1. **Bold** values represent the best method (original and reproduced), whereas *italic* values represent absolute differences greater than 0.05.

	P@10			MAP		
	original	reproduced	diff.	original	reproduced	diff.
BM25	0.4390	0.5048	*−0.0658*	0.2922	0.3052	−0.0130
LM	0.3752	0.4444	*−0.0692*	0.2325	0.2825	−0.0500
CBOW ($\gamma = 0$)	0.1631	0.1635	−0.0004	0.0401	0.0513	−0.0112
CBOW+BM25	0.4610	0.5143	*−0.0533*	0.2986	0.3082	−0.0096
CBOW+LM	0.4438	0.4460	−0.0022	0.2745	0.2828	−0.0083
Yu+BM25	0.4600	0.5143	*−0.0543*	0.2990	0.3112	−0.0122
Online+BM25	**0.4771**	**0.5206**	−0.0435	**0.3005**	0.3078	−0.0073
Yu+LM	0.4467	0.4524	−0.0057	0.2778	0.2772	+0.0006
Online+LM	0.4581	0.4524	+0.0057	0.2793	0.2793	0.0000
Faruqui+BM25	0.4695	0.5143	−0.0448	0.3001	**0.3134**	−0.0133
Offline+BM25	0.4715	0.5143	−0.0428	0.3001	0.3133	−0.0132
Faruqui+LM	0.4470	0.4492	−0.0022	0.2778	0.2822	−0.0044
Offline+LM	0.4486	0.4476	+0.0010	0.2781	0.2823	−0.0042

It is also interesting to note that the performance of our QLM-based re-ranking approaches provide close results to those obtained by Liu et al. In terms of absolute difference, none of

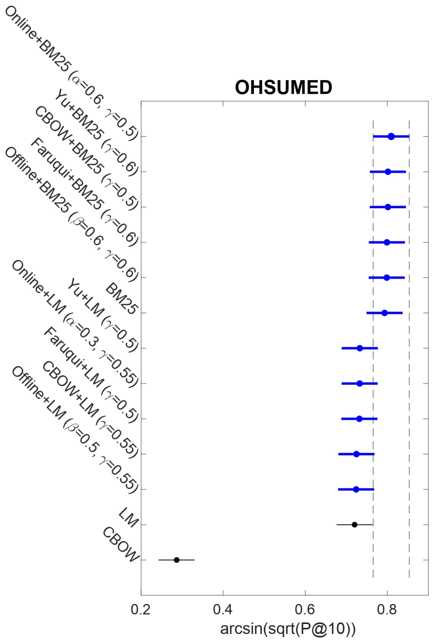


Fig. 6.1 Significance test for the results of the reproduced models reported in Table 6.2. All pairwise comparisons are calculated with Tukey's HSD confidence intervals and a significance level of 0.05. The comparisons are made for P@10.

the QLM-based re-ranking combinations presents a difference greater than 0.01. However, the improvement obtained combining QLM with word embeddings in the reference paper is greater than that obtained in this study – where all the knowledge-enhanced models deteriorate MAP, although they improve P@10.

The results of the Tukey's HSD test in Figure 6.1 confirm the limited improvements of the knowledge-enhanced re-ranking approaches over the other models. In fact, all the models belong to the top group – except for QLM and word2vec when used to perform retrieval over the entire collection (lines two and three of Table 6.2).

The plots in Figure 6.2 show the sensitivity of the α and β hyperparameters for the joint learning and the modified retrofitting models. For each re-ranking combination, values of $\alpha, \beta \in \{0.02, 0.04, 0.1, 0.3, 0.4, 0.5, 0.6, 0.7, 0.9, 1\}$ are tested using the best γ from Table 6.1 and their behavior is compared with the behavior reported in the reference paper. Blue plots represent the behavior of the reproduced models as α/β varies, whereas red plots represent the behavior of the original models.

In general, we observe smaller performance variations for the reproduced versions as α and β change. This is especially true when considering the re-ranking methods that use the joint learning model, where the original versions present performance variations

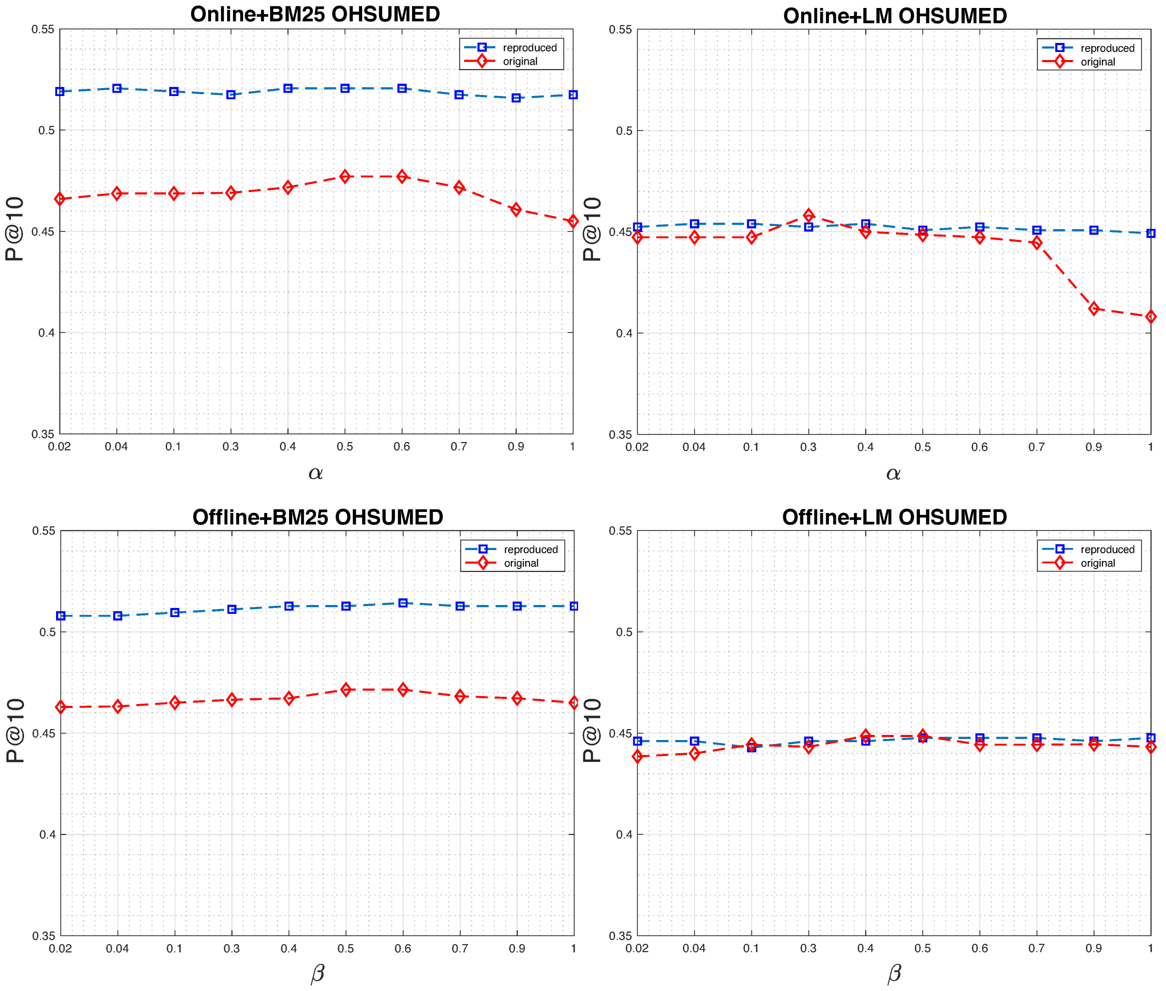


Fig. 6.2 Sensitivity of α and β hyperparameters for the Online and Offline methods, respectively. Blue plots represent the behavior of the reproduced models for P@10 as α/β varies, whereas red plots represent the behavior of the original models. For each re-ranking combination, the hyperparameter γ is selected from Table 6.1.

greater than 0.025 for some α and β values. This indicates that the impact of the relational knowledge, injected into word embeddings during/after the training process, is limited when the embeddings are employed for re-ranking. In fact, relying on lexical models to gather an initial pool of 1000 documents leaves candidate documents most affected by the semantic gap – i.e., relevant documents that do not contain query terms – undiscovered. Consequently, the potential of knowledge-enhanced word embeddings is not fully expressed in a re-ranking scenario.

Finally, we perform a qualitative comparison between the original and the reproduced knowledge-enhanced word embeddings. Table 6.3 reports the 10 most similar words, in terms of cosine similarity, found by the original versions of word2vec, the joint learning model, and the modified retrofitting model respectively. Then, for each word, we report the score obtained in the reference paper, the score obtained with the reproduced versions, and their difference.

Table 6.3 Top 10 most similar words to "Heart" for CBOW, Online, and Offline models respectively. For each model, the first column refers to the top 10 words obtained with the original version (as in [147]), whereas second and third columns refer to the scores obtained by the original and reproduced version, respectively. Symbols ↓, ↑, – mean that the word rank obtained using the reproduced version is lower, higher, or equal, respectively, to that obtained with the original version.

CBOW				Online				Offline			
	original	reproduced	diff.		original	reproduced	diff.		original	reproduced	diff.
Cardiac	0.4891	0.5814 ↓	−0.0923	Cardiac	0.5205	0.9957 –	−0.4752	Cardiac	0.7960	0.6790 –	+0.1170
Synergist	0.4494	0.0281 ↓	+0.4213	Hearts	0.5030	0.9680 ↓	−0.4650	Cor	0.6957	0.3440 ↓	+0.3517
Hearts	0.4276	0.1690 ↓	+0.2586	Cor	0.4939	0.7768 ↓	−0.2829	Synergist	0.5030	0.0898 ↓	+0.4132
Cardiovascular	0.4096	0.3575 ↑	+0.0521	Synergist	0.4690	0.2937 ↓	+0.1753	Hearts	0.4738	0.3773 ↑	+0.0965
Acyanotic	0.3987	−0.0732 ↓	+0.4719	Cardiovascular	0.4156	0.9894 ↑	−0.5738	Biventricular	0.4721	−0.0667 ↓	+0.5388
Ouvrier	0.3934	/	/	Cerebrovascular	0.4149	0.9875 ↑	−0.5726	Cyanotic	0.4720	0.0889 ↓	+0.3831
Multiorgan	0.3931	0.0242 ↑	+0.3689	Acyanotic	0.3985	0.5884 ↓	−0.1899	Cardiorespiratory	0.4714	0.0261 ↓	+0.4453
Ventricular	0.3837	0.6424 ↑	−0.2587	Ventricular	0.3979	0.9946 ↑	−0.5967	Ventricular	0.4651	0.6359 ↑	−0.1708
Cardiorespiratory	0.3829	0.0261 ↑	+0.3568	Cardiorespiratory	0.3969	0.8446 ↑	−0.4477	Acyanotic	0.4585	−0.0178 –	+0.4763
Thrive	0.3766	0.0898 ↑	+0.2868	Biventricular	0.3831	0.7345 ↑	−0.3514	Circulatory	0.4552	0.1917 ↑	+0.2635

We observe that the reproduced version of the joint learning model provides stronger similarities for heart-related words than the original one. Indeed, words like "Cardiac", "Cardiovascular", and "Ventricular" have similarities close to 1. This means that the reproduced model effectively encodes relational knowledge in its word representations. On the other hand, the modified retrofitting model presents results much closer to those obtained with word2vec – although with better similarities for heart-related words. Probably, a higher number of epochs might be required for the modified retrofitting model to effectively incorporate the relational knowledge in the learning process. Thus, compared to the original version, our modified retrofitting model provides weaker results. As a side note, the word "Ouvrier" was not part of our models vocabulary.

6.2.6 The Limitations of Re-Ranking

While reproducing the models used and proposed by Liu et al. [147] on the OHSUMED collection, we encountered many challenges – mainly related to the lack of information regarding the parameters and hyperparameters of the word2vec models. The results of this reproduciblity study showed that, although we successfully reproduced word2vec and confirmed the potential of its knowledge-enhanced extensions, we achieved limited improvements in re-ranking compared to the original results. This might be related to the stronger

BM25 and QLM baselines we obtained. Besides, from the analysis of the sensitivity of the models to different α/β values, presented in Figure 6.2, we observed small performance variations. This suggests that the impact of relational knowledge is limited when the embeddings are used for re-ranking. Thus, the potential of knowledge-enhanced word embeddings might be better expressed in query expansion or rank fusion techniques – where their ability to retrieve candidate documents that are most affected by the semantic gap could be effectively used.

6.3 Knowledge-Enhanced Document Embeddings for IR

Nguyen et al. [169] present two knowledge-enhanced document embedding models: the conceptual doc2vec (cdoc2vec) and the retrofitted doc2vec (rdoc2vec). cdoc2vec considers concepts from an external knowledge resource instead of words when learning document representations. On the other hand, rdoc2vec retrofits document representations learned using words with document representations learned using concepts. The two approaches extend the models of Devine et al. [56] and Faruqui et al. [74], respectively, by introducing document representations in the learning process.

The knowledge-enhanced document embeddings proposed by Nguyen et al. adopt the doc2vec DM architecture. Therefore, we first recall its main aspects and then present the proposed extensions. Given a word vocabulary $V = \{w_1, ..., w_n\}$ and a corpus $D = \{d_1, ...d_m\}$ where each document is defined as an ordered sequence of words, doc2vec DM maximizes the following log-likelihood:

$$\sum_{i=1}^{m} \sum_{w_t \in d_i} \log p(w_t | w_{t-k}^{t+k}, d_i) \tag{6.9}$$

where d_i is the i^{th} document within D, w_t is the t^{th} word within d_i, and w_{t-k}^{t+k} is the set of words within a window of size k centered at w_t (w_t excluded).

6.3.1 The Conceptual Doc2Vec

The cdoc2vec model adopts the doc2vec DM architecture but considers the concept vocabulary C instead of the word vocabulary V. We recall that $C \subseteq \mathscr{C}$ is the set of unique concepts associated with the words contained in the corpus. Therefore, cdoc2vec maximizes the following log-likelihood:

$$\sum_{i=1}^{m} \sum_{c_t \in d_i} \log p(c_t | c_{t-k}^{t+k}, d_i) \tag{6.10}$$

6.3.2 The Retrofitted Doc2Vec

The rdoc2vec model retrofits document representations obtained by doc2vec and cdoc2vec. The objective is to learn document representations that minimize the distance between doc2vec and cdoc2vec representations as follows:

$$\Psi(\boldsymbol{U}) = \sum_{i=1}^{m} \psi(\boldsymbol{u}_i) = \sum_{i=1}^{m} \left[\beta||\boldsymbol{u}_i - \boldsymbol{u}'_i||^2 + (1-\beta)||\boldsymbol{u}_i - \boldsymbol{u}''_i||^2\right] \tag{6.11}$$

where $\boldsymbol{u}_i, \boldsymbol{u}'_i, \boldsymbol{u}''_i \in \mathbb{R}^b$ are the rdoc2vec, doc2vec, and cdoc2vec document representations for all $d_i \in D$, b is the document vectors size, $||\cdot||$ is the euclidean distance, and β a hyperparameter that controls the strength of the word-based and concept-based components.

In [169, p. 165], the authors reported the pseudo-code to solve the optimization problem for learning retrofitted document embeddings. We tested the described optimization process but, unfortunately, we faced convergence issues. In a subsequent personal communication, the authors provided us with the source code they employed to optimize the rdoc2vec model. From that, we derived the revised pseudo-code reported in Algorithm 6.1.

Algorithm 6.1: Retrofitting document vectors

```
Input  :$\boldsymbol{U}' = (\boldsymbol{u}'_{ij}), \boldsymbol{U}'' = (\boldsymbol{u}''_{ij}) \in \mathbb{R}^{m \times b}$
Output :$\boldsymbol{U} = (\boldsymbol{u}_{ij}) \in \mathbb{R}^{m \times b}$
for $i \in \{1,...,m\}$ do
    $\boldsymbol{u}_i \sim \mathscr{U}(-1,1)$
foreach epoch do
    $\Psi(\boldsymbol{U}) = 0$
    for $i \in \{1,...,m\}$ do
        $\Psi(\boldsymbol{U}) \mathrel{+}= \beta||\boldsymbol{u}_i - \boldsymbol{u}'_i||^2 + (1-\beta)||\boldsymbol{u}_i - \boldsymbol{u}''_i||^2$
        $\Delta = 2\beta(\boldsymbol{u}_i - \boldsymbol{u}'_i) + 2(1-\beta)(\boldsymbol{u}_i - \boldsymbol{u}''_i)$
        $\boldsymbol{u}_i = \boldsymbol{u}_i - \alpha\,\Delta$
    $\Psi(\boldsymbol{U}) \mathrel{/}= m$
    if $\Psi(\boldsymbol{U}) < \varepsilon$ then
        break
return $\boldsymbol{U}$
```

The process takes $\boldsymbol{U}'$, $\boldsymbol{U}''$ document representations as inputs and updates for each document d_i its representation $\boldsymbol{u}_i$, using the first derivative $\Delta = \frac{\delta\psi(\boldsymbol{u}_i))}{\delta\boldsymbol{u}_i}$ of $\psi(\cdot)$ with a step size of α. The process stops after a given number of iterations or when $\Psi(\boldsymbol{U})$ reaches a minimum threshold value ε. Note, however, that $\Psi(\cdot)$ is a convex objective function and its optimal (closed-form) solution can be found by computing the first-order derivative and setting it to zero.

6.3.3 Experimental Setup and Implementation Details

Experimental Setup

We reproduce the experiments performed by Nguyen et al. [169] on the OHSUMED collection [107]. We conduct the experiments on the 63 official topics using the *title* field, as indicated by the authors in a personal communication. We do not reproduce the experiments on the TREC Medical collection since it is not available anymore.[5]

Nguyen et al. adopt the MeSH controlled vocabulary as Ω and the Cxtractor tool to perform Entity Linking (EL).[6] Since the authors do not report the MeSH version they used, we based our experiments on the version contained within the UMLS 2018AA release. Note that reporting the exact MeSH version is important to enable reproducibility, since the number of relations and concepts in MeSH changes every year (see Subsection 2.2.2 for more details). Rather than Cxtractor, we use the widely-adopted QuickUMLS [207] to map MeSH-restricted CUIs from UMLS to words within OHSUMED. For each word in V, we rely on QuickUMLS to find, whenever possible, a list of candidate CUIs within UMLS. Then, we restrict the list of CUIs to those belonging to the MeSH controlled vocabulary and we keep the top-ranked occurrence.

The authors propose a PRF based method that relies on word, concept, and document embeddings to perform query expansion. Given a query, a lexical (bag-of-words) model retrieves an initial set of documents. Then, the method leverages the embeddings learned by doc2vec models to match top-ranked (feedback) documents with words or concepts, returning a score for each (word/concept, document) pair. The scores for the pairs that contain the same words/concepts are combined using CombSUM [200] and the top words/concepts are selected to expand the original query. Then, the lexical model retrieves the final set of 1000 documents using the expanded query.

The evaluation measures used in the experiments are: MAP, P@20, and Recall@20. Since doc2vec models are used by the PRF based method to perform query expansion, P@20 is used as the performance indicator for model selection.

[5] https://trec.nist.gov/data/medical.html

[6] https://sourceforge.net/projects/cxtractor/

Implementation Details

Among the challenges we met to reproduce the experiments performed by Nguyen et al. [169], there is the fact that all the doc2vec models considered have been evaluated only through query expansion. Therefore, the only way to verify whether the reproduced doc2vec models are performing as the original ones is by comparing the performances of lexical models when using the expanded query – thus adding more layers to the process. Furthermore, the hyperparameters used to train the doc2vec models – along with the number of top documents, words, and concepts selected for query expansion – are not exhaustively reported. Also, as previously mentioned, we incurred in convergence problems when running the optimization process described in the original paper. In our experiments, the loss value never reached the minimum threshold value ε.

In a personal communication, the authors kindly provided us with the required parameter settings and the code for rdoc2vec and the query expansion method. Below, we report both the data we got from the paper [169] and those obtained thanks to the personal communication (p.c.).

The embeddings size for doc2vec and cdoc2vec models is set to 200 [169]. The other (hyper) parameters are: window size $k = 8$, learning rate equal to 0.02, negative samples set to 5, and minimum word frequency set to 5 (p.c.). Learning rate is decreased linearly during the stochastic gradient descent training process and the Gensim library [183] is used to train doc2vec and cdoc2vec models (p.c.). As preprocessing, non-alphanumeric words are removed (p.c.).[7] There is no information regarding the number of epochs considered to train doc2vec and cdoc2vec. Thus, we train each model for 15 epochs and we select the model iteration that performs best for P@20. The minimum loss for rdoc2vec is set to $\varepsilon = 10^{-7}$, the learning rate to $\alpha = 0.01$, and the optimal $\beta = 0.6$ [169]. The maximum number of training iterations is set to 500 (p.c.).

The lexical model used before and after query expansion is BM25 with default parameters, implemented in Lucene (p.c.).[8] The number of feedback documents considered by the PRF based method is 3, while the number of words/concepts used to expand the query is 5 (p.c.). It is also worth mentioning that the number of words/concepts considered from each feedback document is 100 (p.c.). This means that we do not combine the scores obtained for each word/concept within the vocabulary, but only the scores for the top 100 words/concepts within each feedback document. This leads to very different performances, as can be seen from Tables 6.4 and 6.5.

[7] These information are also reported in a subsequent paper [220] by the same authors.

[8] https://lucene.apache.org/

When performing query expansion with cdoc2vec, the preferred names of the selected CUIs (i.e. concepts) are used as expansion terms (p.c.). On the other hand, the rdoc2vec query expansion relies on words when $\beta \geq 0.5$ and on concepts otherwise (p.c.).

In our experiments, we adopt the (hyper) parameter setting described above but we rely on the ElasticSearch implementation of BM25.

6.3.4 Experimental Results

We report the results of the comparison between the original and the reproduced versions of the doc2vec based query expansions. Table 6.4 reports the results when we select expansion terms by combining the scores of each word/concept within the vocabulary, whereas Table 6.5 reports the results when we combine the scores for the top 100 words/concepts selected from each feedback document.

Table 6.4 Comparison between the original and the reproduced doc2vec based query expansions when we combine the scores of each word/concept within the vocabulary. A positive difference means that the original method does better than the reproduced one. Best models are in **bold**, differences greater than 0.05 are in *italic*.

	MAP		
	orig.	repr.	diff.
doc2vec/PRF	0.1017	**0.0419**	+*0.0598*
cdoc2vec/PRF	0.0956	0.0347	+*0.0609*
rdoc2vec/PRF	**0.1020**	0.0418	+*0.0602*
	P@20		
	orig.	repr.	diff.
doc2vec/PRF	**0.2556**	**0.0508**	+*0.2048*
cdoc2vec/PRF	0.2365	0.0500	+*0.1865*
rdoc2vec/PRF	**0.2556**	0.0500	+*0.2056*
	Recall@20		
	orig.	repr.	diff.
doc2vec/PRF	**0.1086**	**0.0207**	+*0.0879*
cdoc2vec/PRF	0.0980	0.0185	+*0.0795*
rdoc2vec/PRF	**0.1086**	0.0205	+*0.0881*

First of all, we observe that we did not successfully reproduce any of the doc2vec based query expansions. In Table 6.4, the absolute differences between the original and reproduced

Table 6.5 Comparison between the original and the reproduced doc2vec based query expansions when we combine the scores of the top 100 words/concepts selected from each feedback document. A positive difference means that the original method does better than the reproduced one. Best models are in **bold**, differences greater than 0.05 are in *italic*.

	MAP		
	orig.	repr.	diff.
doc2vec/PRF	0.1017	0.0882	+0.0135
cdoc2vec/PRF	0.0956	0.0639	+0.0317
rdoc2vec/PRF	**0.1020**	**0.0883**	+0.0137
	P@20		
	orig.	repr.	diff.
doc2vec/PRF	**0.2556**	0.1143	+*0.1413*
cdoc2vec/PRF	0.2365	0.0810	+*0.1555*
rdoc2vec/PRF	**0.2556**	**0.1151**	+*0.1405*
	Recall@20		
	orig.	repr.	diff.
doc2vec/PRF	**0.1086**	0.0480	+*0.0606*
cdoc2vec/PRF	0.0980	0.0340	+*0.0640*
rdoc2vec/PRF	**0.1086**	**0.0483**	+*0.0603*

versions are all greater than 0.05 and, for P@20, even greater than 0.15. Furthermore, if we consider that BM25 achieves values of 0.0975 for MAP, 0.1444 for P@20, and 0.0598 for Recall@20, we also notice that the reproduced doc2vec based query expansions deteriorate the initial performances.

We observe a similar trend in Table 6.5, where we select expansion terms by combining the scores of the top 100 words/concepts from each feedback document. In terms of MAP, the absolute differences are lower than 0.05 for each doc2vec based query expansion. However, the differences for P@20 and Recall@20 are still greater than 0.05. Again, the reproduced query expansions deteriorate the initial performances.

Thus, the results reported in Tables 6.4 and 6.5 indicate that we did not successfully reproduce the experiments performed by Nguyen et al. [169] – despite relying on the same parameter setup, the code shared by the authors, and the same library to train doc2vec models. Nevertheless, the results we obtained are reasonable if we analyze the performances of the reproduced doc2vec models when they are used to perform retrieval over the entire collection (see Table 6.6). In fact, doc2vec models achieve scores lower than 0.02 in all the considered measures. This means that query and document representations do not effectively match. Therefore, doc2vec models do not find relevant words/concepts for top-ranked (feedback) documents – even though such documents are retrieved by a state-of-the-art IR model – thus introducing detrimental information in the retrieval process.

Table 6.6 Comparison between the doc2vec models when they are used to perform retrieval. For each measure, the best model is in **bold**.

	MAP	P@20	Recall@20
doc2vec	0.0021	0.0119	0.0037
cdoc2vec	**0.0051**	**0.0189**	**0.0092**
rdoc2vec	0.0020	0.0119	0.0037

6.3.5 When Reproducibility Goes Sideways

We attempted to reproduce the experiments performed by Nguyen et al. [169] on the OHSUMED collection. We faced several challenges, mainly related to the lack of information regarding the (hyper) parameters of the doc2vec models, the IR model used to perform retrieval, and the parameters of the PRF based method. The results of this reproducibility study show that, although relying on the same parameter setup, the code shared by the authors, and the same library to train doc2vec models, we did not successfully reproduce the original experiments. We did not find consistent trends with any of the three doc2vec based query expansions considered by Nguyen et al. and all our reproduced versions worsen the performance of BM25.

For a better assessment of the performances, it would have been beneficial knowing the performance of the doc2vec models without the query expansion component. In fact, reporting the individual performances of the neural models adopted in multi-stage IR systems could be a good way to ease reproducibility in challenging situations. Otherwise, it would be even harder for future researchers to understand the possible issues encountered in the implementation of multi-stage IR systems.

6.4 Towards Knowledge-Enhanced Neural Models for IR

The reproducibility study we conducted in Sections 6.2 and 6.3 highlighted some limitations in the way knowledge-enhanced neural language models are applied to IR, as well as in their effectiveness to address the semantic gap between queries and documents. The analysis of the work by Liu et al. [147] in Section 6.2 showed the potentiality of enhancing neural language models through external knowledge. On the other hand, however, the results also underlined the marginal impact that the integration of external knowledge has for re-ranking. In this respect, the sensitivity analysis presented in Figure 6.2 showed that re-ranking performances vary little regardless of the strength of the relational knowledge applied during the learning process. The reason behind this low sensitivity to knowledge integration relates to the fact that, in a re-ranking scenario, knowledge-enhanced neural language models are restricted to the set of candidate documents retrieved by a lexical model. Therefore, they cannot express their full potential and leverage the relational knowledge learned during training, as the set of candidate documents is biased towards lexical signals. As a consequence, relevant documents affected by the semantic gap remain undiscovered.

We reached similar conclusions for the work by Nguyen et al. [169] in Section 6.3 as well, despite the fact we did not successfully reproduce the experiments. From the analysis of the results in Table 6.6, we observed that doc2vec models perform poorly for document retrieval. This further confirms the difference between IR and NLP tasks and highlights the inability of doc2vec models to effectively encode relevant features for IR. Consequently, when used to identify expansion terms from feedback documents for PRF, they fail to provide effective results. Besides, the PRF method proposed by Nguyen et al. [169] relies on lexical models to retrieve the set of feedback documents. Therefore, the poor performances of (knowledge-enhanced) doc2vec models are also exacerbated by the fact that feedback documents are retrieved through lexical matching and are thus biased towards lexical rather than semantic signals. For this reason, we advocate that knowledge-enhanced models should be used at the early stages of the IR pipeline to express their full potential – that is, retrieving feedback documents most affected by the semantic gap. In this way, lexical models could benefit from expansion terms that are more suited to address the semantic gap between queries and documents.

Thus, the outcomes of the reproducibility study highlighted the need for knowledge-enhanced neural IR models capable of providing effective performances at the early stages of the IR pipeline, where the integration of external knowledge can express its full potential. To this end, in the next section, we present our novel unsupervised knowledge-enhanced neural framework for IR, whose representations are optimized for retrieval and encode both polysemy and synonymy to address the semantic gap between queries and documents.

6.5 The Semantic-Aware Neural Framework for IR

We present SAFIR, an unsupervised knowledge-enhanced neural framework for IR. SAFIR jointly learns word, concept, and document representations from scratch and optimizes them for document retrieval. At the same time, SAFIR addresses the semantic gap by modeling polysemy and synonymy. Regarding polysemy, SAFIR contextualizes word meanings by combining word and concept representations in the learning process. Word and concept representations are optimized to minimize the distance between the so combined word meanings and the documents in the vector space. Thus, word meaning representations are created on-the-fly by combining word and concept representations. This compositional process avoids to create a representation for each word meaning, which is an approach prone to data sparsity [40]. On the other hand, SAFIR models synonymy via multi-task learning. Word representations are shared between two learning tasks that are optimized jointly: text matching and word similarity. For the word similarity task, SAFIR minimizes the distance between word representations for words presenting synonymy relations within an external knowledge resource.

6.5.1 Framework

As shown in Figure 6.3, SAFIR has three main components: semantic indexing, representation learning, and semantic matching.

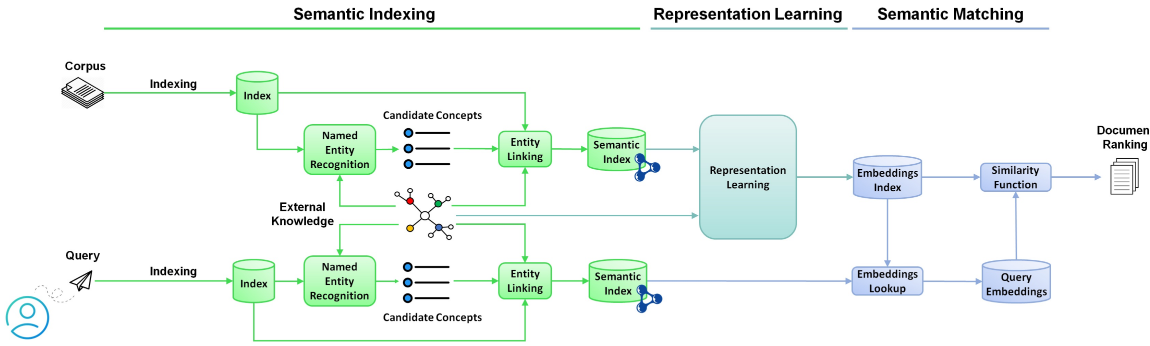


Fig. 6.3 SAFIR overall architecture. The semantic indexing component produces the knowledge-enhanced documents (and queries) along with the required vocabularies. The representation learning component learns word, concept and document representations. Finally, the semantic matching component computes the similarity score between query and document representations and ranks the documents accordingly.

The **semantic indexing** component takes as input a corpus D and a knowledge resource Ω and applies Named Entity Recognition (NER) and Entity Linking (EL) techniques to produce the knowledge-enhanced corpus Φ.

The **representation learning** component relies on the output provided by the semantic indexing component to learn word, concept, and document representations. This component models polysemy and synonymy while optimizing representations for document retrieval.

The **semantic matching** component uses the learned representations to perform semantic matching between a knowledge-enhanced query φ and the documents ϕ. Documents are ranked in decreasing order of the similarity score computed between query and document representations.

6.5.2 Semantic Indexing

We adopt the UMLS metathesaurus as the knowledge resource Ω, and we rely on QuickUMLS [207] to perform NER. We use QuickUMLS to map each word within the word vocabulary V with a list of candidate concepts from UMLS. Given a word, we recall that QuickUMLS relies on approximate matching to compute the similarity between the word and the concept labels within UMLS. Concept labels are terms used by the knowledge resource to express a concept. Thus, candidate concepts are ranked according to the similarity score between a target word and the concept labels. Finally, candidate concepts with a similarity score below a given threshold are pruned from the resulting ranking list.

Algorithm 6.2: Shallow word-sense disambiguation

```
Input    : document $d$, candidate concepts from $d$, and knowledge base $\Omega(\mathscr{C},\mathscr{E})$
Output : knowledge-enhanced document $\phi$
Set of candidate concepts $C_d \leftarrow \emptyset$
foreach word $w \in d$ do
    $C_d \leftarrow C_d \cup C_w$  ($C_w$ : list of candidate concepts associated to $w$ by QuickUMLS)
Output list of word-concept pairs $\phi \leftarrow []$
foreach word $w \in d$ do
    Relative maximum connections $max = 0$
    List of senses associated with $w$, $S_w \leftarrow []$
    foreach candidate concept $\hat{c} \in C_w$ do
        Number of edges $n = |\hat{c}' \in C_d : (\hat{c},\hat{c}') \in \mathscr{E} \wedge \exists\, w' \in d : w' \neq w \wedge \hat{c}' \in C_{w'}|$
        if $n \geq max$ then
            if $n > max$ then
                $S_w \leftarrow [(w,\hat{c})]$
                $max \leftarrow n$
            else
                $S_w \leftarrow S_w \cup [(w,\hat{c})]$
    $(w,c^*) \leftarrow S_w[0]$ (holds $\hat{c}$ ranked highest by QuickUMLS among candidates left)
    $\phi \leftarrow \phi \cup [(w,c^*)]$
return knowledge-enhanced document $\phi$
```

Then, we perform EL over candidate concepts returned by QuickUMLS using our modified version of the Shallow Word Sense Disambiguation (S-WSD) algorithm proposed by Mancini et al. [156]. The modified S-WSD takes as input a document d, the lists of candidate concepts associated with the words in d, and Ω, and it outputs the knowledge-enhanced document ϕ. S-WSD applies to any Ω and has running time linear in the collection size $|D|$. Below, we report the details of our modified version of the S-WSD algorithm. Algorithm 6.2 reports the pseudo-code.

First, we create the set C_d with all the candidate concepts extracted by QuickUMLS for each word $w \in d$ (lines 1 to 3). Secondly, for each candidate concept $\hat{c}$ of w, we compute the number of concepts which are connected with $\hat{c}$ in the knowledge base Ω and are included in C_d, excluding connections of concepts which only appear as candidates of the same word (lines 5 to 9). Finally, each word w is associated with its top candidate concept c^* according to its number of connections in the document. If there are ties, the concept with the highest rank from QuickUMLS is associated with the word. The set of top candidate concepts that are returned by the algorithm forms the concept vocabulary C (lines 10 to 18).

The approach we propose to obtain Φ has two main advantages: (i) it does not require an annotated corpus, which is the biggest bottleneck of supervised EL techniques; (ii) it scales linearly with the corpus size when off-the-shelf disambiguation systems do not [199, 207].

6.5.3 Representation Learning

We develop a shallow neural network learning word, concept, and document representations from scratch. Representations are network parameters in the form of matrices $\{\boldsymbol{w}_i\}_{i=1}^{|V|} \in \mathbb{R}^{|V|\times a}$, $\{\boldsymbol{c}_i\}_{i=1}^{|C|} \in \mathbb{R}^{|C|\times a}$, and $\{\boldsymbol{\phi}_i\}_{i=1}^{|\Phi|} \in \mathbb{R}^{|\Phi|\times b}$ for vocabulary words V, vocabulary concepts C, and knowledge-enhanced documents Φ, respectively, where a denotes the size of word and concept representations and b the size of document representations. The network models polysemy and synonymy while optimizing the representations for retrieval. For polysemy, word and concept representations are composed to generate contextual word meanings at the representation level. Then, the network optimizes sequences of word meanings to be similar to the knowledge-enhanced documents from which they are extracted. This training process approximates query-documents interactions. At the same time, the network constrains the representations of synonyms to be similar to each other. Therefore, we can divide the network into three main parts: **polysemy modeling**, **retrieval modeling**, and **synonymy modeling**. Figure 6.4 depicts the general architecture of the representation learning component.

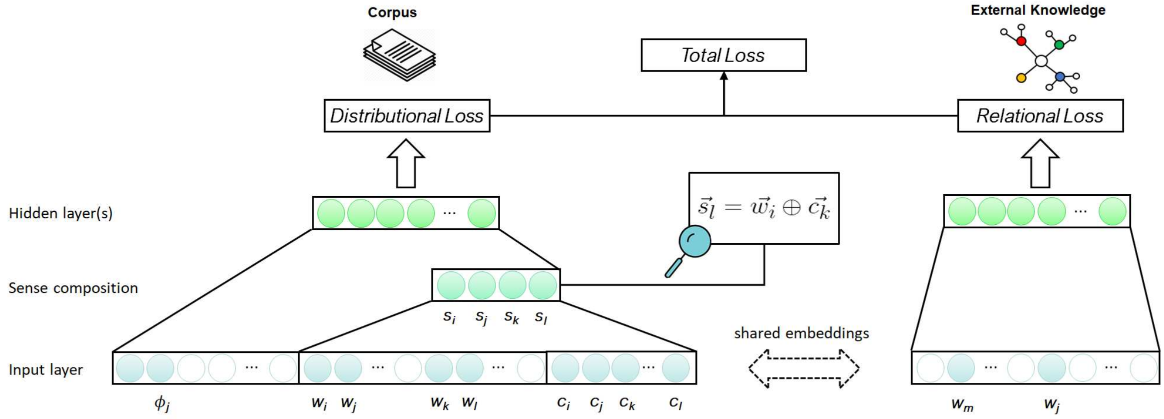


Fig. 6.4 Neural architecture of the representation learning component. Distributional loss minimizes the distance between document and (contextual) word-concept representations, whereas relational loss minimizes the distance between word representations for words presenting synonymy relations within the knowledge resource.

Polysemy Modeling: The network performs a word sense composition process to integrate polysemy. The network models the representation of each word-concept pair $\langle w, c\rangle$ as the element-wise sum of its word and concept representations via a compositional function f

$$\boldsymbol{s} = f(\boldsymbol{w}, \boldsymbol{c}) = \boldsymbol{w} \oplus \boldsymbol{c} \tag{6.12}$$

whose output $\boldsymbol{s}$ is the contextual representation of the word-concept pair $\langle w, c\rangle$. Thus, word meanings are defined in the vector space through a translation process from the word $\boldsymbol{w}$ to its contextual meaning $\boldsymbol{s}$ by the concept $\boldsymbol{c}$, i.e., $\boldsymbol{c}$ acts as a translation vector. In other words, given a word w and, say, two concepts c_1 and c_2 associated with w in different contexts, the compositional function $f(\cdot,\cdot)$ outputs different representations depending on the concept – and thus the context – considered. Therefore, polysemous words obtain distinct representations according to the context where they appear.

In this way, all the possible combinations of contextual representations are generated on-the-fly offline (training) or online (retrieval). This avoids the need for a word sense vocabulary. Learning representations based on a word sense vocabulary is prone to data sparsity and can lead to underfitting the representations of rare word meanings [40].

Then, the network employs the contextual representations to learn matching relations for retrieval modeling. Matching relations are learned together with synonymy relations via multi-task learning. Word representations are shared between two learning tasks that are optimized jointly: text matching and word similarity.

Retrieval Modeling: We adopt neural vector space models [223] for text matching. A neural vector space model takes as input a batch $\mathscr{B}$ of document/sequence pairs and minimizes the distance between their representations. We define a sequence of size k sampled from ϕ and starting at position h as $S_h^k(\phi) = (\langle w_j, c_j \rangle)_{j=h}^{h+k-1}$. Then, the representation of the input sequence $S_h^k(\phi)$ is defined as the average of its word-concept pair representations:

$$\boldsymbol{S}_h^k(\phi) = \frac{1}{k} \sum_{i=h}^{h+k-1} f(\boldsymbol{w}_i, \boldsymbol{c}_i) = \frac{1}{k} \sum_{i=h}^{h+k-1} \boldsymbol{s}_i \tag{6.13}$$

where the word-concept representations $\boldsymbol{s}_i$ are computed as in (6.12). Then, L2-normalization is applied to the sequence representation $\boldsymbol{S}_h^k(\phi)$, followed by a linear transformation:

$$\boldsymbol{h}_h^k(\phi) = \boldsymbol{W} \cdot \text{norm}(\boldsymbol{S}_h^k(\phi)) \tag{6.14}$$

where $\boldsymbol{W} \in \mathbb{R}^{b \times a}$ is a projection matrix. The L2 norm$(\cdot)$ function makes the feature values proportionate to each other. Since the objective of the text matching task is to minimize the distance between a document ϕ and a sequence $S_h^k(\phi)$ sampled from it, this means that during training the network learns to prioritize some word-concept representations over others when minimizing the distance between a document and a sequence sampled from it. From an IR perspective, the network learns to boost the representation of word-concept pairs that are discriminative for the target document. On the other hand, the linear transformation forces the sequence representation to encode the aspects relevant for text matching into the document space. The network optimizes the projection matrix $\boldsymbol{W}$ to transfer relevant aspects of the sequence representation from the word-concept space $\mathbb{R}^a$ to the document space $\mathbb{R}^b$. Basically, norm$(\cdot)$ boosts the representation of discriminative word-concept pairs and $\boldsymbol{W}$ projects relevant aspects of the sequence representation into the document space.

Before computing the similarity between a sequence $S_h^k(\phi)$ and a document ϕ, batch normalization [120] is applied to the input sequences, followed by the hard-tanh$(\cdot)$ activation function:

$$\overline{\boldsymbol{h}}_h^k(\phi) = \text{hard-tanh}(\text{batch-norm}(\boldsymbol{h}_h^k(\phi), B)) \tag{6.15}$$

Batch normalization reduces the internal covariate shift and hard-tanh$(\cdot)$ introduces linear behavior around zero to allow gradients to flow easily when the unit is not saturated, while providing a clear decision in the saturated regime [93].

Thus, the similarity between a document ϕ and a sequence $S_h^k(\phi)$ is defined as:

$$P(y|\phi, S_h^k(\phi)) = \sigma(\boldsymbol{\phi} \cdot \overline{\boldsymbol{h}}_h^k(\phi)) \tag{6.16}$$

where $\overline{\boldsymbol{h}}_h^k(\phi)$ is the standardized representation of the input sequence, $\sigma(\cdot)$ is the sigmoid function, and y is a binary random variable equal to one if $S_h^k(\phi)$ belongs to ϕ and zero otherwise.

An adjusted-for-bias variant of the NCE loss [98] is used to train the network for the text matching task. NCE maximizes the similarity between the representations of the document ϕ and the sequence $S_h^k(\phi)$ sampled from it, while it minimizes the similarity between $S_h^k(\phi)$ and t contrastive documents – i.e., documents not containing the sequence. The reweighting scheme applied to NCE removes the dependence on the number of contrastive documents t, since large values of t bias the network towards contrastive documents. This training procedure mimics query-documents interactions. The log-probability of a document ϕ given the sequence $S_h^k(\phi)$ is defined as:

$$\log \overline{P}(\phi|S_h^k(\phi)) = \frac{t+1}{2t}\Big(t \log P(y|\phi, S_h^k(\phi)) + \sum_{\substack{i=1,\\ \phi_i\sim\mathscr{U}(\Phi)}}^{t} \log(1.0 - P(y|\phi_i, S_h^k(\phi)))\Big) \quad (6.17)$$

where $\mathscr{U}(\Phi)$ represents the uniform distribution over documents Φ used to obtain the t contrastive examples. Therefore, the loss function used to optimize the network for the text matching task, averaged over the batch size $|B|$, is:

$$L_{\text{dis}}(\Theta|B) = -\frac{1}{|B|}\sum_{i=1}^{|B|} \log \overline{P}(\phi_i|S_h^k(\phi_i)) \quad (6.18)$$

where Θ is the set of parameters $\big\{\{\boldsymbol{w}_i\}_{i=1}^{|V|}, \{\boldsymbol{c}_i\}_{i=1}^{|C|}, \{\boldsymbol{\phi}_i\}_{i=1}^{|\Phi|}, \boldsymbol{W}\big\}$. We refer to this loss as the distributional loss, since it relies on the distributional hypothesis [102].

Synonymy Modeling: To integrate synonymy, the network relies on the set of synonym pairs $R = \{\langle\langle w_i, c_k\rangle, \langle w_j, c_k\rangle\rangle \mid w_i \neq w_j \wedge c_k \in C\}$ of the corpus Φ and performs word similarity. The objective of the word similarity task is to minimize the distance between two words that are synonyms in Ω. Hence, the network optimizes the representations for words expressing the same concept to be close in the vector space. We define the similarity between two synonyms as:

$$P(y|\langle\langle w_i, c\rangle, \langle w_j, c\rangle\rangle) = \sigma(\boldsymbol{w}_i \cdot \boldsymbol{w}_j) \quad (6.19)$$

where y is a binary random variable equal to one if both w_i and w_j express c and zero otherwise.

Then, the loss function used to optimize the network for the word similarity task, averaged over the batch size $|B|$, is:

$$L_{\text{rel}}(\Theta|R) = -\frac{1}{|B|} \sum_{\langle\langle w_i,c\rangle,\langle w_j,c\rangle\rangle \in R} \log P(y|\langle\langle w_i,c\rangle,\langle w_j,c\rangle\rangle) \tag{6.20}$$

where we recall that $R = \{\langle\langle w_i,c\rangle,\langle w_j,c\rangle\rangle \mid w_i \neq w_j \wedge c \in C\}$ is the set of synonym pairs of the corpus Φ. We refer to this loss as the relational loss, as it relies on the relational constraints provided by Ω.

The relational loss presents similarities with the constrained embeddings proposed by Liu et al. [147] (see Subsection 6.2.2). Compared to that, the relational loss we employ acts as a regularizer which keeps minimizing the distance between words that are synonyms as the training for the text matching task progresses. On the other hand, our approach differs from retrofitting [74] since it is performed during training and not as a second stage of learning (see Subsection 6.2.3). By modeling synonymy as a second-stage regularization, we would end up modifying word representations that have already been optimized towards text matching. In this way, word-concept and document representations could misalign and the network might lose effectiveness on text matching, which is the main task.

Finally, we apply L2 regularization over Θ parameters:

$$L_{\text{reg}}(\Theta) = \frac{1}{2|B|}\Big(\sum_{i=1}^{|V|} ||\boldsymbol{w}_i||_2^2 + \sum_{j=1}^{|C|} ||\boldsymbol{c}_j||_2^2 + \sum_{k=1}^{|\Phi|} ||\boldsymbol{\phi}_k||_2^2 + ||\boldsymbol{W}||_F^2\Big) \tag{6.21}$$

L2 regularization enforces the network to use all its parameters without depending too heavily on any of them. Therefore, the loss function used to train the entire network is the combination of the L2 regularization and the loss functions for the text matching and word similarity tasks:

$$L(\Theta|B,R) = L_{\text{dis}}(\Theta|B) + \lambda \cdot L_{\text{rel}}(\Theta|R) + \gamma \cdot L_{\text{reg}}(\Theta) \tag{6.22}$$

where λ controls the extent to which synonyms representations are brought close during training and γ controls the regularization strength. Parameters Θ are optimized using Adam [128], an adaptive learning rate optimization function. Adam updates every parameter with every batch. This means that parameters are updated even when they have a zero gradient. Hence, Adam dampens the consequences of applying the hard-tanh activation function, which leads to zero gradients in the saturated regime.

Thus, the network learns contextual representations for word-concept pairs (polysemy) that are close to the corresponding document representations (retrieval) and, at the same time, to synonym representations (synonymy).

6.5.4 Semantic Matching

The learned representations $\{\boldsymbol{w}_i\}_{i=1}^{|V|} \in \mathbb{R}^{|V| \times a}$, $\{\boldsymbol{c}_i\}_{i=1}^{|C|} \in \mathbb{R}^{|C| \times a}$, and $\{\boldsymbol{\phi}_i\}_{i=1}^{|\Phi|} \in \mathbb{R}^{|\Phi| \times b}$ are then used to perform semantic matching between query and document representations. We define the representation of a query φ as follows:

$$\boldsymbol{\varphi} = \boldsymbol{W} \cdot \frac{1}{n} \sum_{i=1}^{n} f(\boldsymbol{w}_i, \boldsymbol{c}_i) \tag{6.23}$$

We treat the query similarly to a sequence by first averaging its word-concept representations and then projecting it into the document space through $\boldsymbol{W}$, which is the projection matrix learned during training. Finally, the matching score between the query φ and a document ϕ is given by the cosine similarity between their representations $(\boldsymbol{\varphi}, \boldsymbol{\phi})$ in the document space $\mathbb{R}^b$.

6.6 Experimental Setup

6.6.1 Test Collections and Knowledge Resource

We consider four standard collections for medical literature retrieval: OHSUMED [107], TREC Clinical Decision Support 2014 (CDS14) [188], 2015 (CDS15) [189], and 2016 (CDS16) [184]. For OHSUMED, we use the *description* query field and we perform experiments on the 63 official topics. For CDS collections, we use the *summary* query field. Statistics for each collection are reported in Table 6.7. Further details on the considered test collections can be found in Section 2.1. As knowledge resource, we adopt the 2018AA release of the UMLS metathesaurus [29].

Table 6.7 Statistics for the OHSUMED, CDS14, CDS15, and CDS16 collections. Arithmetic mean and standard deviation are reported for document and query lengths.

	OHSUMED	CDS14	CDS15	CDS16
Collection				
Document Count	348,566	733,138	733,138	1,255,259
Vocabulary	294,520	663,528	663,528	852,739
Document Length	95.82±62.85	117.56±107.16	117.56±107.16	121.76±142.87
Queries				
Query Count	63	30	30	30
Query Length	7.05±3.00	25.63±9.24	20.87 ± 6.55	32.83 ± 17.29

6.6.2 Evaluation Measures and Statistical Tests

We use nDCG@1000, nDCG@100, nDCG@10, P@10, and Recall@1000 to evaluate models. We also consider infNDCG for CDS collections, since it is the reference measure adopted in the TREC CDS tracks. infNDCG cannot be computed for the OHSUMED collection as inferred measures require sampled relevance judgments not available for OHSUMED.

We perform the post-hoc Tukey's HSD test [221] with one-way ANOVA to test statistical significance. Again, we apply the Tague-Sutcliffe transformation to Tukey's HSD tests [218].

6.6.3 Retrieval Strategies

We consider two retrieval strategies to investigate the research questions: document retrieval and query expansion.

Document Retrieval is the typical retrieval strategy where systems retrieve a set of candidate documents given a query. Documents are ranked according to the similarity score computed between them and the query.

Query Expansion consists in expanding the original query with additional terms that can help systems to retrieve more relevant documents. Query expansion addresses the semantic gap by using expansion terms to retrieve relevant documents that do not necessarily match the original query. We rely on RM3 [137, 121], an effective PRF based method which typically achieves good retrieval performance at the cost of executing an additional round of retrieval. The set of ranked documents R_1 from the first round of retrieval is used to select expansion terms to augment the query for the second round of retrieval. Formally, the RM3 query language model is defined as:

$$\begin{aligned} P(w|\mathrm{RM3}) &= \alpha P(w|q) + (1-\alpha) \sum_{d \in R_1} P(w|d) P(d|q) \\ &= \alpha \frac{tf(w,q)}{|q|} + (1-\alpha) \sum_{d \in R_1} \frac{tf(w,d)}{|d|} \mathrm{sim}(q,d) \end{aligned} \tag{6.24}$$

where w is a potential expansion term, α is an interpolation hyperparameter, $tf(\cdot,\cdot)$ is the term frequency, and $\mathrm{sim}(\cdot,\cdot)$ is the similarity function of the model used in the first round of retrieval.

6.6.4 Semantic Indexing Setup

We preprocess the document collections using Whoosh. The preprocessing comprises tokenization and stopwords removal. We rely on the Indri stoplist [215] for stopwords removal. The preprocessed collections are then indexed using Gensim [183]. We index title and abstract fields. This limits noise injection in the training of knowledge-enhanced representation models. Besides, article abstracts from medical literature often present a rich and structured nature that suits to IR tasks and helps us to validate our research questions [31].
For NER and EL, we consider UMLS concepts from the default semantic types provided by QuickUMLS, as they are typically associated with the four aspects of the medical decision criteria: symptoms, diagnostic tests, diagnoses, and treatments. As suggested by Limsopatham et al. [143], these semantic types represent the necessary information health practicioners need to assist their patients. Regarding QuickUMLS, we set the similarity threshold to the default value of 0.7.

Semantic index statistics are presented in Table 6.8. Table 6.8(a) shows statistics for the number of candidate concepts per word identified by QuickUMLS (NER), whereas Table 6.8(b) shows statistics for the number of synonyms per concept after the use of S-WSD (EL). Then, knowledge-enhanced collection statistics are presented in Table 6.9. Table 6.9(a-b) show statistics for the number of concepts per document and query, while Table 6.9(c-d) show statistics for the number of polysemous words per document and query. Finally, statistics for the S-WSD algorithm are reported in Table 6.10, where statistics are counted over all documents (or queries) and consider the subset of words with at least two candidate concepts associated. In particular, non-disambiguated words refer to those words for which the S-WSD algorithm does not prune the initial list of candidate concepts provided by QuickUMLS.

Table 6.8 Semantic index statistics for: (a) number of candidate concepts per word, (b) number of synonyms per concept. Statistics are computed for the subset of words/concepts belonging to the term/concept dictionary of SAFIR – therefore they represent a fraction of the collection statistics. (a) considers only words with at least one candidate concept associated (i.e., roughly the 20% of the term dictionary in each collection).

	(a) concepts/word				(b) synonyms/concept			
	OHSUMED	CDS14	CDS15	CDS16	OHSUMED	CDS14	CDS15	CDS16
Max	67	67	67	67	25	35	35	29
Min	1	1	1	1	1	1	1	1
Median	1	1	1	1	1	1	1	1
Average	1.85	1.77	1.77	1.77	1.84	1.78	1.78	1.78
Std Dev	1.96	1.94	1.94	1.87	1.63	1.64	1.64	1.62

Table 6.9 Knowledge-enhanced collection statistics for: (a) number of concepts per document, (b) number of concepts per query, (c) number of polysemous words per document, and (d) number of polysemous words per query.

	(a) concepts/document				(b) concepts/query			
	OHSUMED	CDS14	CDS15	CDS16	OHSUMED	CDS14	CDS15	CDS16
Max	308	14934	14934	39016	9	25	14	38
Min	0	0	0	0	1	3	3	3
Median	54	62	62	65	4	10	10	11
Average	53.74	61.29	61.29	63.61	4.05	11.10	9.13	13.57
Std Dev	36.01	57.74	57.74	77.01	1.63	4.81	3.19	8.31
	(c) polysemy/document				(d) polysemy/query			
	OHSUMED	CDS14	CDS15	CDS16	OHSUMED	CDS14	CDS15	CDS16
Max	187	7164	7164	18438	7	16	11	22
Min	0	0	0	0	0	1	1	2
Median	29	31	31	32	3	7	6	8
Average	30.98	31.75	31.75	33.13	2.78	7.20	6.17	9.07
Std Dev	21.69	30.94	30.94	40.48	1.51	3.54	2.62	5.26

Table 6.10 S-WSD statistics for: % of disambiguated words by S-WSD, % of non-disambiguated words by S-WSD, and execution time of S-WSD. Statistics are counted over all documents/queries and consider the subset of words with at least two candidate concepts associated.

	% disambiguated words	% non-disambiguated words	Exec. time (sec)
Collection			
OHSUMED	47.84	52.16	3,872
CDS14	41.49	58.51	8,360
CDS15	41.49	58.51	8,360
CDS16	41.64	58.36	15,938
Queries			
OHSUMED	52.00	48.00	–
CDS14	47.22	52.78	–
CDS15	52.43	47.57	–
CDS16	44.49	55.51	–

6.6.5 Retrieval Models Setup

We consider three categories of retrieval models: lexical (bag-of-words) models, corpus-driven models, and knowledge-enhanced models. As Bag-of-Words (BoW) models we consider:

(1) BM25 [192] with $k_1 = 1.2$ and $b = 0.75$.

(2) Query Likelihood Model (QLM) [254] with Dirichlet smoothing $\mu = 2000$.

As Corpus-Driven (CD) models we consider only those used by knowledge-enhanced models as part of their learning process, that is:

(3) word2vec [162, 235] with skip-gram architecture, where query and document representations are constructed by summing up the representation of the words contained in them. For document representations, we sum word representations weighted by the term IDF [190] as in [147].

(4) doc2vec [138] with DBOW architecture. The query representation is the sum of its word representations.

(5) Neural Vector Space Model (NVSM) [223]. In NVSM, the query representation is the average of its word representations projected to the document space.

As Knowledge-Enhanced (KE) models we consider:

(6) retrofitted word2vec (rword2vec) [74, 147] with $\alpha_i = 1$ and $\beta_i = \text{degree}(i)^{-1}$, where i is the node the update is applied to. We use rword2vec to retrofit word2vec embeddings.

(7) conceptual doc2vec (cdoc2vec) [169] with DBOW architecture. cdoc2vec is trained over the knowledge-enhanced corpus Φ relying only on the concept vocabulary C. The query representation is the sum of its concept representations.

(8) retrofitted doc2vec (rdoc2vec) [169, 220] retrofits document embeddings from doc2vec and cdoc2vec models. We derive the loss function to obtain the optimal (closed-form) solution. The weighting factor β is optimized in the $(0,1)$ range with sweep 0.1. The query representation is the sum of its word representations when $\beta \geq 0.5$ and of its concept representations otherwise.

(9) Semantic-Aware Neural Framework for IR (SAFIR) with word/concept representation size $a = 300$, document representation size $b = 256$, number of contrastive documents $t = 10$, learning rate $\eta = 0.001$, regularization weight $\gamma = 0.001$, synonymy strength λ optimized in the $(0,1]$ range with sweep 0.1, and batch size $|B| = 51200$. We consider three variants of SAFIR: SAFIR$_{\text{sp}}$, which integrates both synonymy and polysemy; SAFIR$_{\text{s}}$ which integrates synonymy but not polysemy (i.e., it takes words only as input); and SAFIR$_{\text{p}}$ which integrates polysemy but not synonymy (i.e., it does not consider the word similarity task).

We rely on Elasticsearch to implement BM25 and QLM. For word2vec, doc2vec, and cdoc2vec models we use Gensim, where we disable vocabulary filtering and frequent word sub-sampling to keep the input consistent in all representation models. We set the embedding size to 256, the number of contrastive examples to 10, and the learning rate $\eta = 0.025$ with linear decay $\eta_{\text{min}} = 0.0001$. We set the sequence size of SAFIR and the two-sided window size of neural language models to 16. For NVSM, we disable both the contextual representations (polysemy) and the word similarity task (synonymy) and we set the remaining parameters as for SAFIR. For corpus-driven and knowledge-enhanced models, the word vocabulary size is limited to the 2^{17} most frequent words that have a document frequency greater than 1 and lower than or equal to $\frac{|\Phi|}{2}$. For knowledge-enhanced models, we rely on the 2018AA release of the UMLS metathesaurus.

SAFIR, NVSM, word2vec, doc2vec, and cdoc2vec are trained for 15 iterations. For each model, we select the iteration that performs best in terms of nDCG@1000, for OHSUMED, and infNDCG for CDS collections. rword2vec and rdoc2vec retrofit optimized word2vec and doc2vec/cdoc2vec, respectively. rword2vec is trained for 10 iterations as the procedure converges to changes lower than 10^{-2} in the Euclidean distance [74]. For SAFIR$_{\text{sp}}$ and

SAFIR$_\text{s}$, we obtain optimal values of the synonymy strength hyperparameter λ equal to 0.1 for both variants in all CDS collections, whereas we obtain values of λ equal to 1.0 and 0.8 for SAFIR$_\text{sp}$ and SAFIR$_\text{s}$, respectively, in OHSUMED.

We select the best iteration to evaluate models based on their top performance for the reference measure. On the other hand, the reader can find details on the performances of SAFIR averaged over iterations 10-15 in Appendix A, where we compare it with NVSM and BM25/RM3 for document retrieval. We also report the behavior of SAFIR variants in terms of optimization as training progresses. Then, we perform Kendall's τ correlations between the rankings of the models obtained when we take the best iteration and the average of iterations 10-15. In this way, we can understand to what extent the ranking of the considered models changes when we consider the average of iterations 10-15 instead of the best iteration. The results show that in more than 60% of cases correlation is greater than or equal to 0.80 – which indicates that the differences between rankings do not reflect noticeable changes [231]. The rest of the correlation values divides among 0.60 (13% of cases), 0.40 (17% of cases), and 0.20 (9% of cases). Correlation values of 0.60 occur with two swaps between models in the ranking list, whereas scores of 0.40 and 0.20 with three and four swaps, respectively. Furthermore, low correlations (i.e., 0.40 and 0.20) cluster on precision-oriented measures, which are highly sensitive to changes across iterations. For these measures, SAFIR$_\text{s}$ and SAFIR$_\text{sp}$ change rank in most collections and BM25/RM3 gains positions. More details can be found in Appendix A.

Lexical models are considered to see how they deal with the relevant documents most affected by the semantic gap (**RQ1**). Corpus-driven models are considered as a basis for comparison to evaluate the ability of knowledge-enhanced models to integrate external knowledge in the learning process (**RQ2**). Knowledge-enhanced baselines are compared with SAFIR to investigate both **RQ1** and **RQ2**. Furthermore, the three variants of SAFIR are compared to each other to understand which linguistic feature impacts retrieval the most (**RQ1**) and how knowledge resources are better used to bridge the semantic gap between query and documents (**RQ2**).

6.6.6 Expansion Models Setup

We adopt lexical, corpus-driven, and knowledge-enhanced models to perform the first round of retrieval, whereas we use only lexical models for the second round. We use equation (6.24) to interpolate query terms with the top m expansion terms from the set of ranked documents R_1. Depending on the model considered, $\text{sim}(\cdot,\cdot)$ is computed between regular q and d representations or their knowledge-enhanced φ and ϕ versions. We adopt the models optimized for document retrieval and we keep Indri default values for RM3, that is $R_1 = 10$, $m = 10$, and $\alpha = 0.5$.[9]

We consider different categories of retrieval models in the first round of retrieval to evaluate their effectiveness in reducing the semantic gap. Precisely, we investigate whether models that are specifically designed to address the semantic gap retrieve relevant documents that lexical models fail to discover. Our intuition is that semantic models – by retrieving relevant documents different from lexical models – allow RM3 to select expansion terms that are more effective in reducing the semantic gap, thus improving the effectiveness of lexical models in the second round of retrieval. Furthermore, we compare corpus-driven and knowledge-enhanced models to analyze how different linguistic features impact on the choice of expansion terms (**RQ1**) and if knowledge-enhanced models are best suited to this retrieval strategy (**RQ2**).

6.7 Document Retrieval: Experimental Results

We present the experimental results for document retrieval and we discuss them based on the research questions. Table 6.11 shows model performances for document retrieval. In addition to the retrieval models reported above, we also consider BM25/RM3 as a lexical baseline. Even though RM3 does not explicitly model polysemy nor synonymy, it is an effective PRF based method that addresses the semantic gap. By considering BM25/RM3 in this evaluation, we can thus investigate differences and similarities between traditional PRF based methods and SAFIR.

6.7.1 The Impact of Polysemy and Synonymy on Document Retrieval

RQ1 Which feature between synonymy and polysemy can be exploited to reduce the semantic gap and improve retrieval?

We see that all SAFIR variants belong to the top performing group (†) for all measures in all the considered collections. This indicates that SAFIR effectively encodes the text matching signals required to perform retrieval regardless of the linguistic feature(s) modeled. Among the three variants, SAFIR$_\text{p}$ provides the best results in CDS collections for most measures. Regarding OHSUMED, SAFIR$_\text{sp}$ is the top performing variant – closely followed by SAFIR$_\text{p}$ – for all measures but Recall@1000, where SAFIR$_\text{s}$ achieves the highest score.

In CDS collections, SAFIR$_\text{s}$ and SAFIR$_\text{sp}$ exhibit performances close to or slightly lower than those of NVSM and SAFIR$_\text{p}$, respectively. We identify two reasons for this.

[9] https://sourceforge.net/p/lemur/code/HEAD/tree/indri/tags/release-5.16/src/RMExpander.cpp

Table 6.11 Retrieval performances of considered models. Models are grouped by type: Bag-of-Words (BoW), Corpus-Driven (CD), Knowledge-Enhanced (KE), and SAFIR. In CDS collections, models are optimized by infNDCG, whereas in the OHSUMED collection models are optimized by nDCG@1000. **Bold** values represent the highest scores among the models in each collection. † represents the models belonging to the statistical top group for the given collection with $\alpha \leq 0.05$.

		infNDCG				nDCG@1000			
		CDS14	CDS15	CDS16	OHSUMED	CDS14	CDS15	CDS16	OHSUMED
BoW	QLM	0.1015†	0.1277†	0.1204†	–	0.1750	0.1577	0.1568†	0.5552†
	BM25	0.1064†	0.1276†	0.1399†	–	0.1838†	0.1579	0.1643†	0.5875†
	BM25/RM3	0.1384†	**0.1578**†	**0.1688**†	–	0.2316†	0.2183†	**0.2068**†	**0.6253**†
CD	word2vec	0.0954†	0.1159†	0.0928	–	0.1548	0.1634†	0.1054	0.5902†
	doc2vec	0.0242	0.0302	0.0292	–	0.0414	0.0453	0.0239	0.3082
	NVSM	0.1576†	0.1449†	0.1475†	–	0.2649†	0.2213†	0.1818†	0.5977†
KE	rword2vec	0.0896†	0.1142†	0.0790	–	0.1501	0.1589†	0.0980	0.5852†
	cdoc2vec	0.0317	0.0517	0.0324	–	0.0430	0.0721	0.0335	0.2330
	rdoc2vec	0.0327	0.0513	0.0292	–	0.0429	0.0718	0.0248	0.2067
SAFIR	SAFIR$_\text{s}$	0.1602†	0.1498†	0.1546†	–	0.2546†	0.2240†	0.1783†	0.6046†
	SAFIR$_\text{p}$	**0.1608**†	0.1516†	0.1523†	–	**0.2723**†	0.2247†	0.1858†	0.6106†
	SAFIR$_\text{sp}$	0.1566†	0.1515†	0.1599†	–	0.2651†	**0.2266**†	0.1849†	0.6144†

		nDCG@100				nDCG@10			
		CDS14	CDS15	CDS16	OHSUMED	CDS14	CDS15	CDS16	OHSUMED
BoW	QLM	0.1035†	0.1207†	0.1013†	0.3974†	0.1384†	0.2013†	0.1150†	0.3736†
	BM25	0.1098†	0.1233†	0.1078†	0.4392†	0.1530†	**0.2166**†	**0.1606**†	0.4429†
	BM25/RM3	0.1338†	**0.1522**†	**0.1298**†	**0.4746**†	0.1645†	0.1986†	0.1518†	0.4618†
CD	word2vec	0.0821	0.1064†	0.0619	0.4461†	0.1028	0.1435†	0.0977†	**0.4754**†
	doc2vec	0.0209	0.0242	0.0196	0.1915	0.0327	0.0211	0.0368	0.1915
	NVSM	0.1362†	0.1385†	0.1077†	0.4181†	0.1694†	0.1664†	0.1324†	0.3873†
KE	rword2vec	0.0774	0.1032†	0.0590	0.4421†	0.0967†	0.1410†	0.0930†	0.4709†
	cdoc2vec	0.0215	0.0454	0.0178	0.1355	0.0317	0.0547	0.0225	0.1165
	rdoc2vec	0.0213	0.0452	0.0202	0.1114	0.0293	0.0588	0.0397	0.0916
SAFIR	SAFIR$_\text{s}$	0.1385†	0.1411†	0.1071†	0.4216†	0.1729†	0.1818†	0.1374†	0.4121†
	SAFIR$_\text{p}$	**0.1435**†	0.1395†	0.1113†	0.4361†	**0.1931**†	0.2053†	0.1519†	0.4267†
	SAFIR$_\text{sp}$	0.1401†	0.1403†	0.1098†	0.4397†	0.1898†	0.1926†	0.1475†	0.4380†

		P@10				Recall@1000			
		CDS14	CDS15	CDS16	OHSUMED	CDS14	CDS15	CDS16	OHSUMED
BoW	QLM	0.1400†	0.2233†	0.1600†	0.4381†	0.2375	0.1836	0.2289†	0.7964†
	BM25	0.1667†	0.2600†	**0.2167**†	0.5016†	0.2503†	0.1826	0.2286†	0.7973†
	BM25/RM3	0.1833†	0.2433†	0.2067†	**0.5413**†	0.3151†	0.2884†	**0.3059**†	0.8431†
CD	word2vec	0.1133	0.1900†	0.1167†	0.5048†	0.2200	0.2194	0.1515	0.7778
	doc2vec	0.0367	0.0367	0.0267	0.2190	0.0660	0.0671	0.0305	0.4795
	NVSM	0.2033†	0.2333†	0.1600†	0.4333	0.3833†	0.3093†	0.2617†	**0.8584**†
KE	rword2vec	0.1267†	0.1967†	0.1133	0.5048†	0.2221	0.2151	0.1414	0.7672
	cdoc2vec	0.0433	0.0933	0.0233	0.1476	0.0658	0.1017	0.0555	0.3889
	rdoc2vec	0.0367	0.1033	0.0333	0.1175	0.0651	0.1018	0.0313	0.3601
SAFIR	SAFIR$_\text{s}$	0.1967†	0.2267†	0.1733†	0.4619†	0.3607†	**0.3134**†	0.2545†	0.8582†
	SAFIR$_\text{p}$	**0.2333**†	**0.2633**†	0.1700†	0.4762†	**0.3846**†	0.3098†	0.2782†	0.8548†
	SAFIR$_\text{sp}$	0.2200†	0.2467†	0.1633†	0.4794†	0.3733†	0.3110†	0.2747†	0.8520†

First, NVSM/SAFIR$_\mathrm{s}$ and SAFIR$_\mathrm{p}$/SAFIR$_\mathrm{sp}$ pairs share the same input data, that is words (the former) and word-concept pairs (the latter). Secondly, the optimal values for the hyperparameter λ that controls the synonymy strength are equal to 0.1 for both variants in all CDS collections. This suggests that the impact of synonymy in CDS collections might be limited or even detrimental. In particular, we expect that modeling polysemy helps to order relevant documents in top positions of the ranking list, while modeling synonymy helps to retrieve a higher number of relevant documents which contain synonyms of the query terms. While the results confirm this trend for polysemy, they do not for synonymy. In fact, both SAFIR$_\mathrm{s}$ and SAFIR$_\mathrm{sp}$ achieve higher results than NVSM and SAFIR$_\mathrm{p}$, respectively, for Recall@1000 and nDCG@1000 in CDS15 only. The negative results of rword2vec – which models synonymy – compared to those of word2vec further support this intuition. On the other hand, cdoc2vec – which addresses both synonymy and polysemy by learning representations over documents composed only of concepts – achieves better results than doc2vec for most measures. Therefore, the results suggest that polysemy impacts more than synonymy on retrieval performances for CDS collections.

Regarding lexical baselines, all SAFIR variants achieve better performances than QLM and BM25 for most measures in all CDS collections. In particular, for nDCG@1000 and Recall@1000, SAFIR variants statistically outperform QLM in CDS14 and both QLM and BM25 in CDS15. On the contrary, BM25/RM3, by performing an additional round of retrieval to expand the original query, improves BM25 performances for most measures. Even though the differences between SAFIR variants and BM25/RM3 are not statistically significant, BM25/RM3 achieves performances greater than SAFIR$_\mathrm{s}$ and SAFIR$_\mathrm{sp}$ for several cases of the measures considered. Conversely, SAFIR$_\mathrm{p}$ outperforms BM25/RM3 for the considered measures more than 60% of the time. Interestingly, BM25/RM3 fails to improve BM25 for precision-oriented measures in CDS15 and CDS16. In both collections, SAFIR$_\mathrm{p}$ outperforms BM25/RM3 for nDCG@10. This suggests that RM3 might fail to answer semantically hard queries that require to handle polysemy – which reinforces our hypothesis on the impact of polysemy in CDS collections.

For reference purposes, we report the best values obtained during TREC CDS tracks for infNDCG – which is the reference measure adopted in these tracks. In CDS14, the best score for infNDCG is 0.2674 [188]. In CDS15, the best score is 0.2939 [189]. Finally, the best score in CDS16 is 0.2815 [184]. Compared to the results in Table 6.11, the scores achieved by the best systems submitted to TREC CDS tracks are higher. Note that these systems rely on a variety of different IR components, ranging from pre- and post-retrieval query expansions to re-ranking, and other components, like classifiers. On the other hand, in this evaluation, we exclusively focus on retrieval models and their ability to retrieve relevant documents most affected by the semantic gap.

Compared to CDS collections, the results on OHSUMED show a lower gap among the models considered. Lexical models, SAFIR variants, NVSM, and word2vec models behave similarly. The only notable exceptions are for P@10, where NVSM does not belong to the top group (†), and Recall@1000, where word2vec models are statistically outperformed by lexical models, NVSM, and SAFIR variants. Our intuition is that the short, keyword-based nature of OHSUMED queries and the limited corpus vocabulary (see Table 6.7) impact on the effectiveness of modeling polysemy and synonymy. Besides, short, keyword-based queries favor models relying on corpus-based features. This explains the competitive performances of lexical and word2vec models – which exploit explicit feature engineering by relying on IDF. In particular, word2vec is the top performing system for nDCG@10. Nevertheless, the results of SAFIR$_{\mathrm{sp}}$ show that polysemy and synonymy can be effectively modeled together. Among the three variants, SAFIR$_{\mathrm{sp}}$ achieves the best results for nDCG@10, nDCG@100, and nDCG@1000, which indicate its ability to retrieve a higher number of relevant documents (synonymy) and to order them in top positions of the ranking list (polysemy). Furthermore, the differences between NVSM/SAFIR$_{\mathrm{s}}$ and SAFIR$_{\mathrm{p}}$/SAFIR$_{\mathrm{sp}}$ pairs favor SAFIR variants integrating synonymy. Also, the optimal values for the hyperparamter λ that controls the synonymy strength are equal to 0.8 and 1.0 for SAFIR$_{\mathrm{s}}$ and SAFIR$_{\mathrm{sp}}$, respectively. This indicates the greater impact that modeling synonymy has for OHSUMED rather than CDS.

The performance of BM25/RM3 further confirms the effectiveness of lexical models in OHSUMED. In particular, BM25/RM3 outperforms all the considered models for most measures. The only exceptions are nDCG@10 and Recall@1000, where word2vec and SAFIR/NVSM achieve higher scores, respectively.

To further investigate the impact of polysemy and synonymy in the considered collections, we perform quantitative and qualitative analyses. Quantitative analyses investigate the degree of polysemy and synonymy within collections, whereas qualitative analyses focus on semantically hard queries that require to handle polysemy and/or synonymy.

Polysemy Analysis

We rely on knowledge-enhanced collection statistics (see Table 6.9) to identify the degree of polysemy within documents and queries. Then, we perform a qualitative analysis between SAFIR variants to evaluate the impact that the integration of polysemy has in ordering relevant documents in top positions of the ranking list. For each collection, we compute the per-topic differences between SAFIR variants in terms of nDCG@10. We rely on Figures 6.5–6.8 to present and discuss the results. The outcomes of the qualitative analysis are used for a second analysis, where we compare SAFIR variants and BM25/RM3 on semantically hard queries. The objective is to understand whether the effectiveness of SAFIR_{p} and SAFIR_{sp} on highly polysemous queries holds against BM25/RM3. To this end, we compute the per-topic differences between SAFIR variants and BM25/RM3 in terms of nDCG@10. Figures 6.9–6.12 complement the analysis.

The Degree of Polysemy. When we compare the average number of words per document from Table 6.7 and the average number of concepts per document from Table 6.9(a) we observe that the average number of concepts is about half the average number of words in all collections. Furthermore, Table 6.9(c) shows that, on average, more than half of the words presenting concepts are polysemous. Similar observations can also be made for queries, where the average number of concepts per query fluctuates between one third and a half of the average number of words depending on the collection – as indicated in Table 6.9(b). Besides, Table 6.9(d) shows that in all collections more than 60% (on average) of the query words presenting concepts are polysemous. These results indicate the large presence of polysemy within the considered collections.

The Impact of Modeling Polysemy. Figures 6.5–6.8 point out an interesting behavior of the different SAFIR variants on single queries. Each figure shows the per-topic differences between SAFIR variants at nDCG@10 for a given collection. The green (light) stems indicate that the upper-side SAFIR variant achieves a higher nDCG@10 score than the lower-side variant. Vice versa, red (dark) stems indicate that the lower-side SAFIR variant achieves a higher nDCG@10 score than the upper-side variant. Overall, the nDCG@10 results are not consistently in favor of one or the other SAFIR variant. Depending on the query, a particular SAFIR variant outperforms the other and vice versa. Nevertheless, SAFIR_{p} and SAFIR_{sp} show results closer to each other than to SAFIR_{s} since they both model polysemy.

In OHSUMED (Figure 6.5), SAFIR_{p} and SAFIR_{s} achieve higher nDCG@10 scores for thirty-one and twenty-nine queries, respectively, whereas on three queries they perform equally. Overall, SAFIR_{p} achieves a higher nDCG@10 score than SAFIR_{s} in more queries (with per-topic difference ≥ 0.10). In particular, SAFIR_{p} outperforms SAFIR_{s} by a large margin (≥ 0.30) on topic OHSU14. If we analyze the degree of polysemy of topic OHSU14, we find out that 50% (two out of four) of the query words are polysemous. Thus, polysemy has a strong impact on this query and SAFIR_{p} (but also SAFIR_{sp}) effectively captures it. A similar trend is found for topic OHSU7, where 75% (three out of four) of the words are polysemous. The smaller difference between SAFIR_{s} and SAFIR_{p}, along with the fact that SAFIR_{sp} achieves the best results among the three variants, suggest that both polysemy

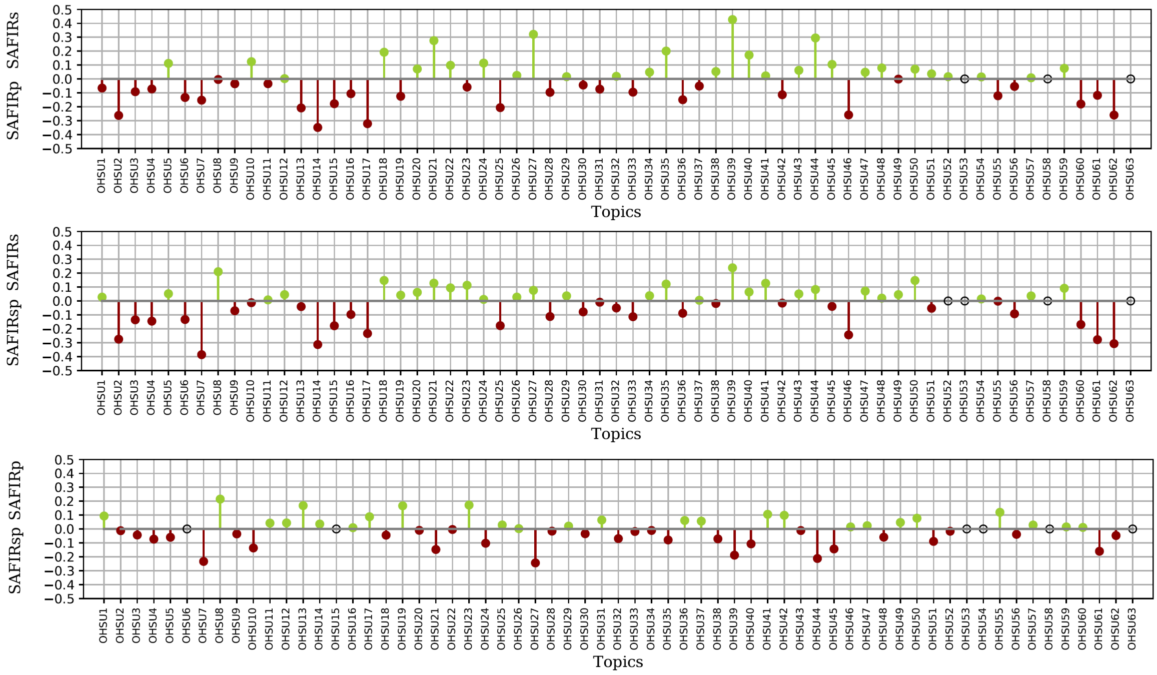


Fig. 6.5 Per-topic differences between SAFIR variants at nDCG@10 in OHSUMED collection. The green (light) stems indicate that the upper-side SAFIR variant achieves a higher nDCG@10 score than the lower-side variant. Vice versa, red (dark) stems indicate that the lower-side SAFIR variant achieves a higher nDCG@10 score than the upper-side variant.

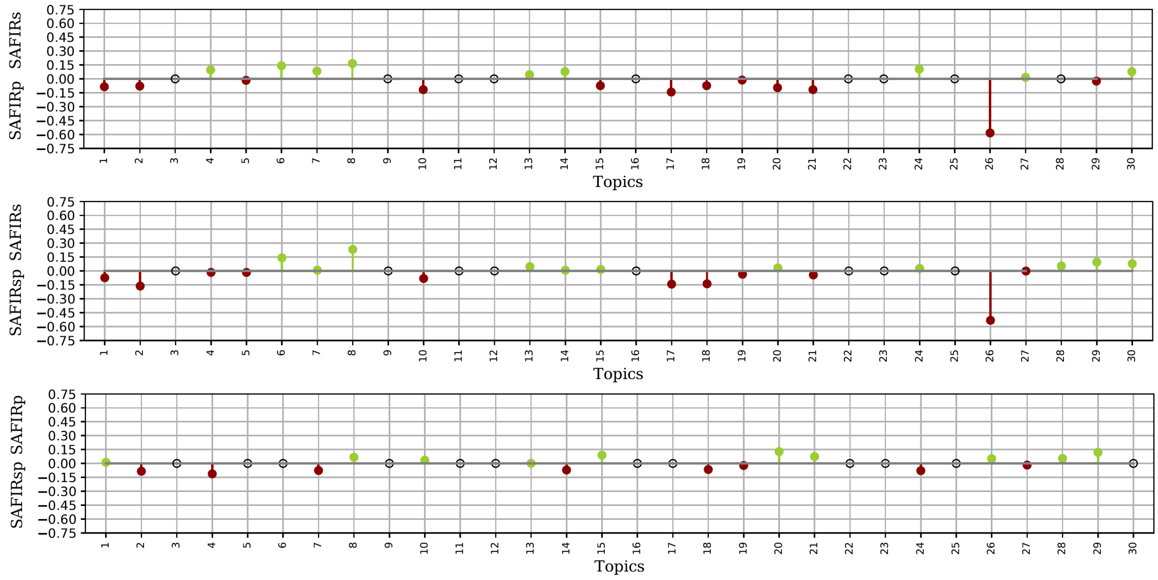


Fig. 6.6 Per-topic differences between SAFIR variants at nDCG@10 in CDS14 collection. The green (light) stems indicate that the upper-side SAFIR variant achieves a higher nDCG@10 score than the lower-side variant. Vice versa, red (dark) stems indicate that the lower-side SAFIR variant achieves a higher nDCG@10 score than the upper-side variant.

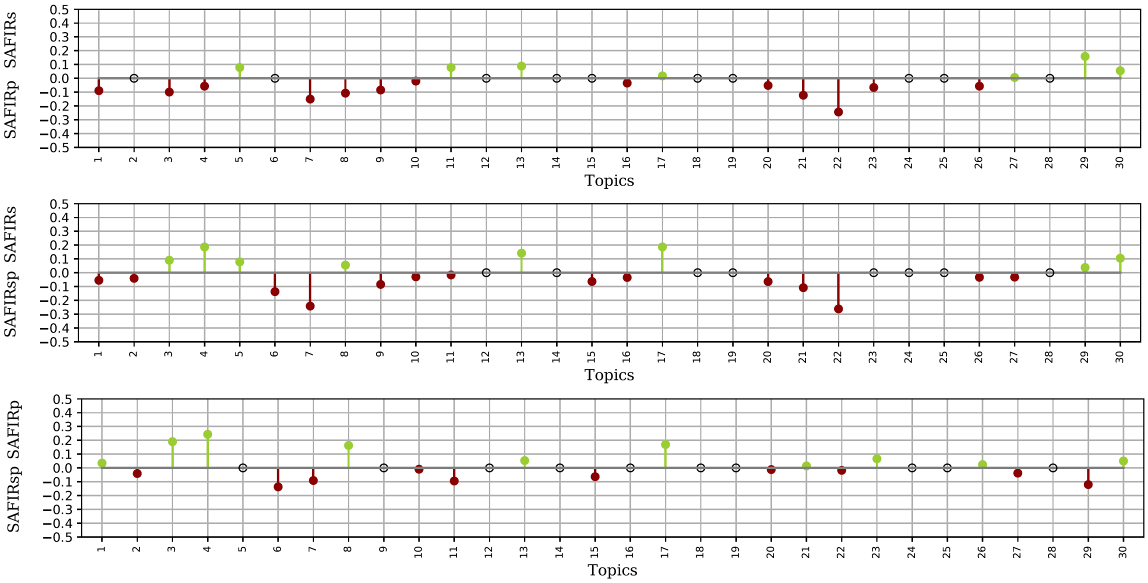


Fig. 6.7 Per-topic differences between SAFIR variants at nDCG@10 in CDS15 collection. The green (light) stems indicate that the upper-side SAFIR variant achieves a higher nDCG@10 score than the lower-side variant. Vice versa, red (dark) stems indicate that the lower-side SAFIR variant achieves a higher nDCG@10 score than the upper-side variant.

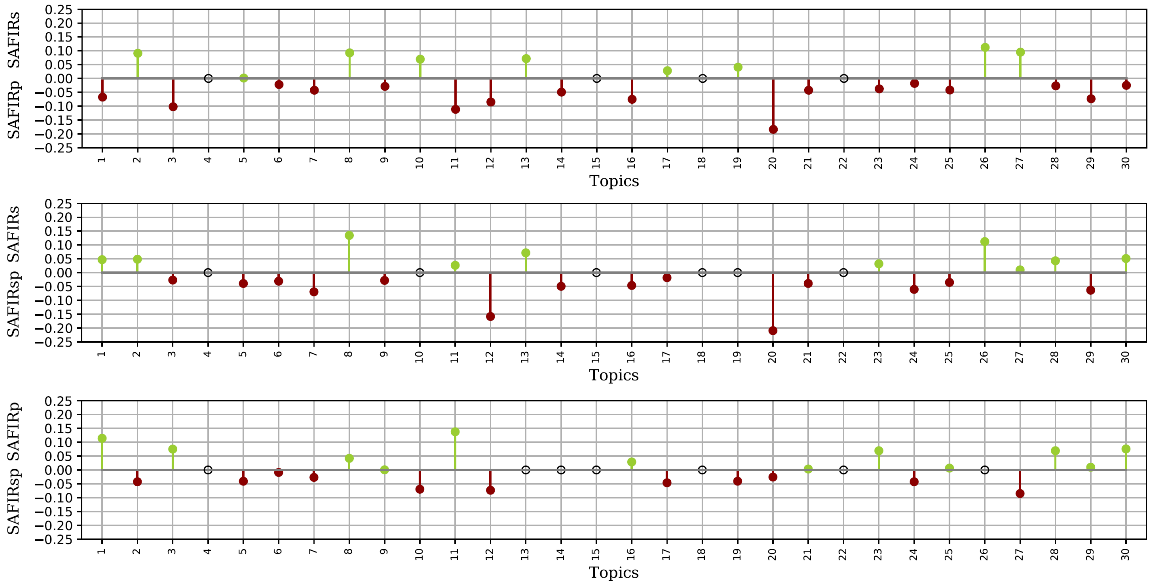


Fig. 6.8 Per-topic differences between SAFIR variants at nDCG@10 in CDS16 collection. The green (light) stems indicate that the upper-side SAFIR variant achieves a higher nDCG@10 score than the lower-side variant. Vice versa, red (dark) stems indicate that the lower-side SAFIR variant achieves a higher nDCG@10 score than the upper-side variant.

and synonymy impact on this query. Conversely, topic OHSU39 presents three polysemous words out of nine (30%). In this case, modeling polysemy impacts less on – or even harms – the query and other factors dominate the performances. We hypothesize that these factors are related to synonymy given the high performance of SAFIR$_{\mathrm{s}}$ and the fact that SAFIR$_{\mathrm{sp}}$ outperforms SAFIR$_{\mathrm{p}}$.

In CDS14 (Figure 6.6), the results show a similar trend to OHSUMED. However, the differences between variants are smaller (with per-topic differences ≤ 0.15). The only notable exception is topic 26, where both SAFIR$_{\mathrm{p}}$ and SAFIR$_{\mathrm{sp}}$ outperform SAFIR$_{\mathrm{s}}$ by a large margin. In this query, 50% of the words are polysemous. Also, the fact that SAFIR$_{\mathrm{p}}$ and SAFIR$_{\mathrm{sp}}$ achieve nearly the same nDCG@10 score suggests that polysemy dominates performances on this query. Conversely, in topic 6, where SAFIR$_{\mathrm{s}}$ outperforms both SAFIR$_{\mathrm{p}}$ and SAFIR$_{\mathrm{sp}}$, less than 30% of the words are polysemous. For this query, both SAFIR$_{\mathrm{p}}$ and SAFIR$_{\mathrm{sp}}$ achieve an nDCG@10 score of zero – which means that polysemy hurts performance even when jointly modeled with synonymy, as in SAFIR$_{\mathrm{sp}}$.

The differences between SAFIR variants are smaller in CDS15 (Figure 6.7), where the largest difference between SAFIR$_{\mathrm{s}}$ and the variants integrating polysemy is found for topic 22 with a value close to 0.30. Also this query presents a large number of polysemous words (50%). Again, SAFIR$_{\mathrm{p}}$ and SAFIR$_{\mathrm{sp}}$ achieve nearly the same nDCG@10 score. The fact that the difference is in favor of SAFIR$_{\mathrm{sp}}$ indicates that the combination of both synonymy and polysemy is beneficial for this query.

Regarding CDS16 (Figure 6.8), the results are in line with those from CDS14 and CDS15. The query presenting the largest difference is topic 20, where SAFIR$_{\mathrm{p}}$ and SAFIR$_{\mathrm{sp}}$ outperform SAFIR$_{\mathrm{s}}$ by a margin of 0.20. In this case, however, the number of polysemous words is lower than in the previous examples, with a percentage of polysemous words of 40%. As for topic 22 from CDS15, the difference between SAFIR$_{\mathrm{p}}$ and SAFIR$_{\mathrm{sp}}$ is in favor of SAFIR$_{\mathrm{sp}}$. Finally, we highlight topic 26, where both SAFIR$_{\mathrm{p}}$ and SAFIR$_{\mathrm{sp}}$ achieve a score of zero for nDCG@10 – as opposed to SAFIR$_{\mathrm{s}}$. Interestingly, topic 26 is an outlier in terms of query length, with a total of fifty-four words of which twenty-two are polysemous. The results for this query show that integrating synonymy is effective, whereas polysemy harms performances.

Thus, the analysis shows that SAFIR$_{\mathrm{p}}$ and – to a lesser extent – SAFIR$_{\mathrm{sp}}$ present a larger number of queries than SAFIR$_{\mathrm{s}}$, where they achieve higher scores for nDCG@10. In particular, when the degree of polysemy within queries is high, SAFIR$_{\mathrm{p}}$ and SAFIR$_{\mathrm{sp}}$ effectively capture it and get high results for precision-oriented measures. On the other hand, SAFIR$_{\mathrm{p}}$ and SAFIR$_{\mathrm{sp}}$ are outperformed by SAFIR$_{\mathrm{s}}$ in some queries where the polysemy degree is low. In such cases, modeling synonymy is effective as opposed to polysemy – which leads to detrimental effects on performances.

The Advantage of Modeling Polysemy. Figures 6.9–6.12 highlight a behavior similar to the one found in the previous analysis. Each figure shows the per-topic differences between SAFIR variants and BM25/RM3 at nDCG@10 for a given collection. The green (light) stems indicate that the SAFIR variant achieves a higher nDCG@10 score than BM25/RM3. Vice versa, red (dark) stems indicate that BM25/RM3 achieves a higher nDCG@10 score than the SAFIR variant. Overall, the nDCG@10 results tend to split between SAFIR variants and BM25/RM3. However, the objective of this analysis is to verify whether the effectiveness of SAFIR$_p$ and SAFIR$_{sp}$ on highly polysemous queries holds against BM25/RM3. Therefore, we compare SAFIR variants with BM25/RM3 on the same highly polysemous queries discussed in the previous analysis.

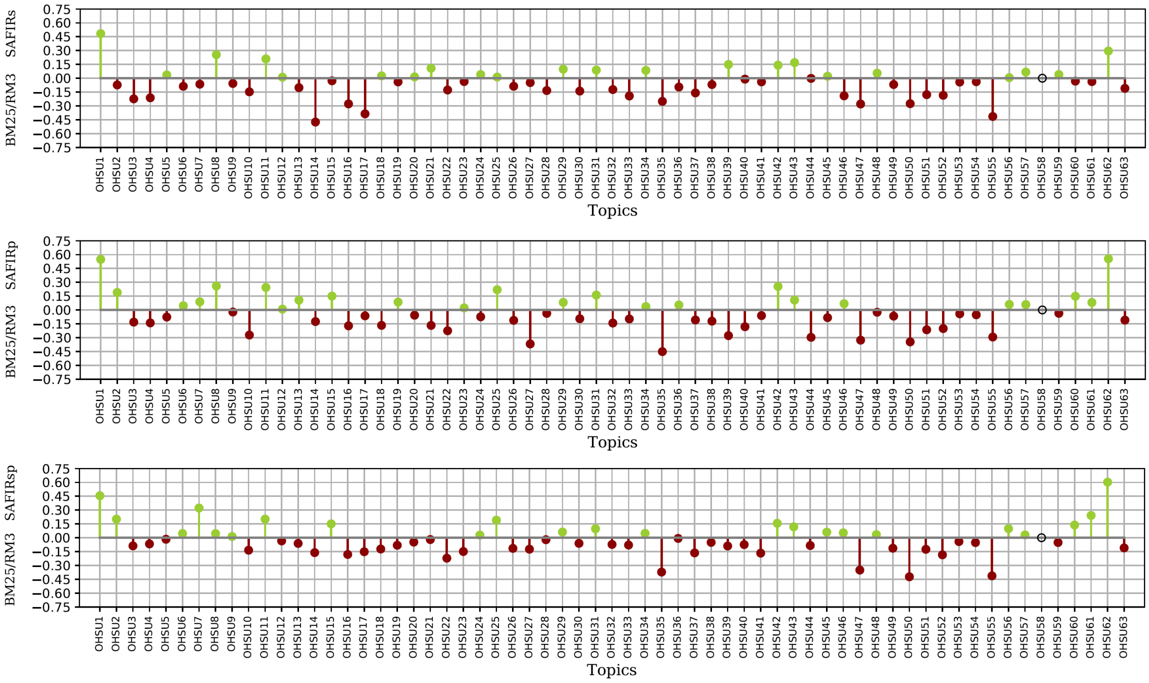


Fig. 6.9 Per-topic differences between SAFIR variants and BM25/RM3 for nDCG@10 in OHSUMED collection. The green (light) stems indicate that the SAFIR variant achieves a higher nDCG@10 score than BM25/RM3. Vice versa, red (dark) stems indicate that BM25/RM3 achieves a higher nDCG@10 score than the SAFIR variant.

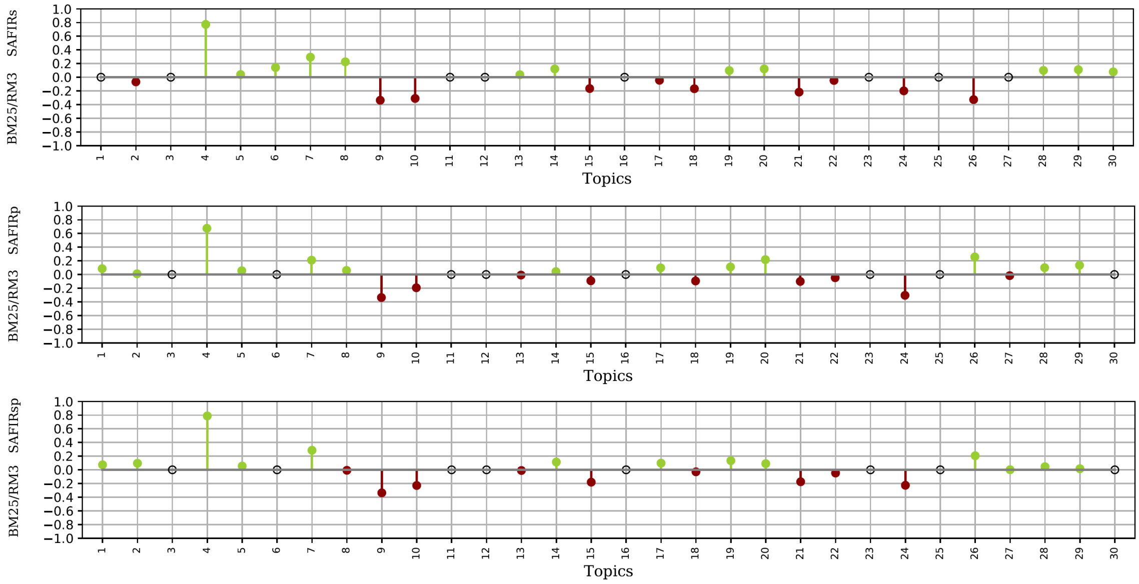


Fig. 6.10 Per-topic differences between SAFIR variants and BM25/RM3 for nDCG@10 in CDS14 collection. The green (light) stems indicate that the SAFIR variant achieves a higher nDCG@10 score than BM25/RM3. Vice versa, red (dark) stems indicate that BM25/RM3 achieves a higher nDCG@10 score than the SAFIR variant.

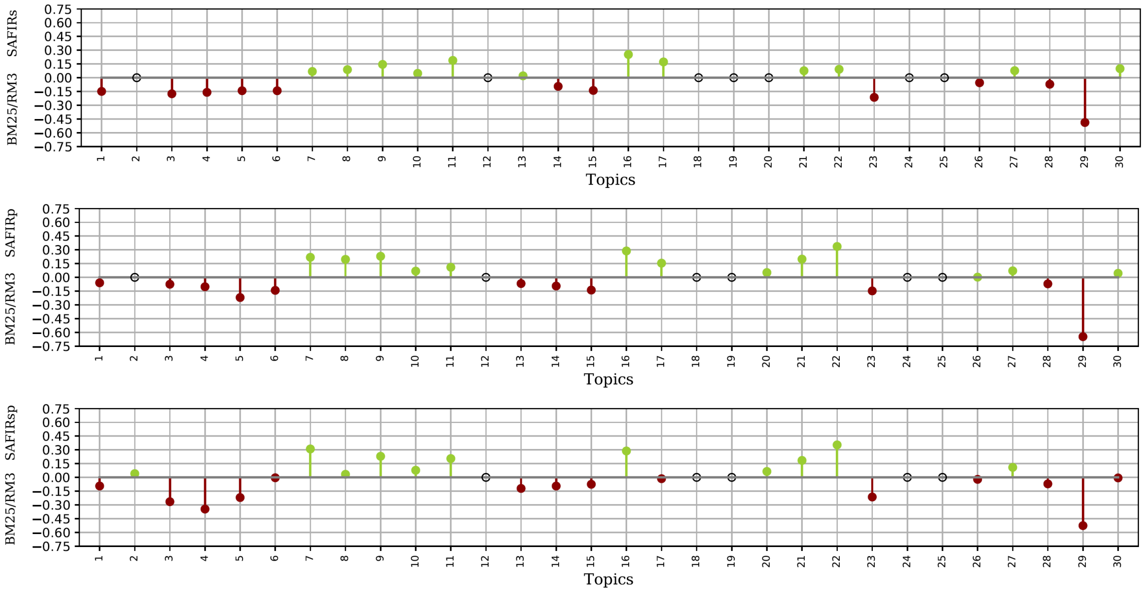


Fig. 6.11 Per-topic differences between SAFIR variants and BM25/RM3 for nDCG@10 in CDS15 collection. The green (light) stems indicate that the SAFIR variant achieves a higher nDCG@10 score than BM25/RM3. Vice versa, red (dark) stems indicate that BM25/RM3 achieves a higher nDCG@10 score than the SAFIR variant.

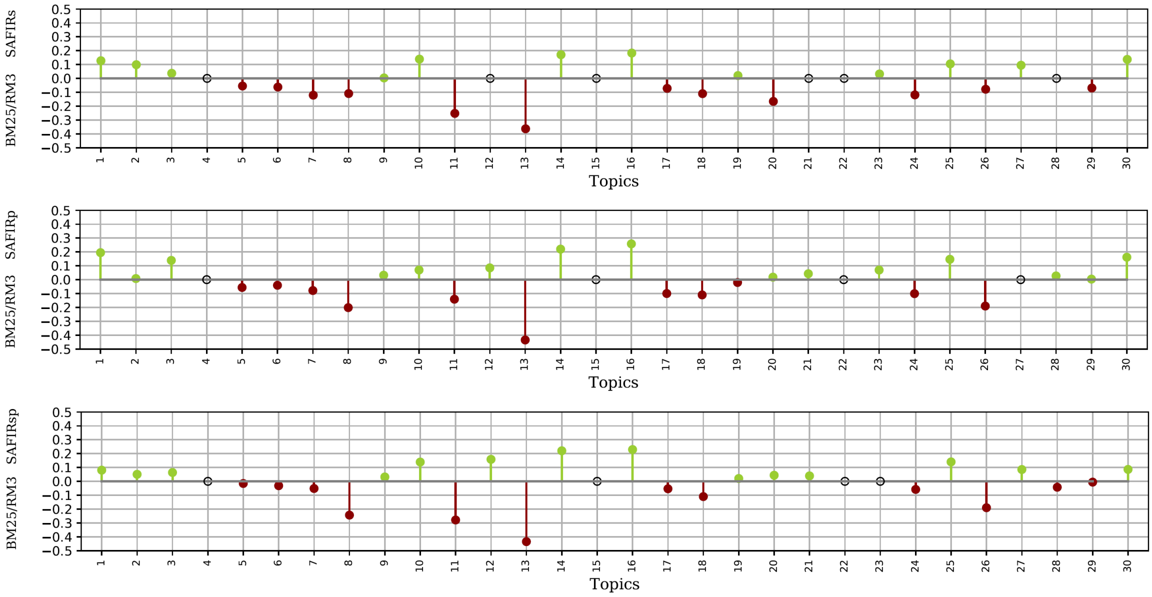


Fig. 6.12 Per-topic differences between SAFIR variants and BM25/RM3 for nDCG@10 in CDS16 collection. The green (light) stems indicate that the SAFIR variant achieves a higher nDCG@10 score than BM25/RM3. Vice versa, red (dark) stems indicate that BM25/RM3 achieves a higher nDCG@10 score than the SAFIR variant.

Regarding OHSUMED (Figure 6.9), in topic OHSU14 – where 50% of the words are polysemous – neither SAFIR$_\mathrm{p}$ nor SAFIR$_\mathrm{sp}$ outperform BM25/RM3. However, both SAFIR variants present smaller differences (about 0.15) with BM25/RM3 if compared to SAFIR$_\mathrm{s}$ (> 0.45). Therefore, although not entirely, modeling polysemy helps SAFIR to bridge the performance gap with BM25/RM3. On the other hand, the results for topic OHSU7 – where 75% of the words are polysemous – show that SAFIR$_\mathrm{p}$ and SAFIR$_\mathrm{sp}$ outperform BM25/RM3. Besides, the fact that SAFIR$_\mathrm{s}$ performs worse than BM25/RM3 indicates that this topic highly benefits from modeling polysemy.

In CDS14 (Figure 6.10), topic 26 (50% of polysemous words) shows a similar trend to topic OHSU7. Also in this case, SAFIR$_\mathrm{p}$ and SAFIR$_\mathrm{sp}$ outperform BM25/RM3 – with per-topic difference ≥ 0.20 – whereas SAFIR$_\mathrm{s}$ does not achieve competitive performance – with a gap of almost 0.40 with BM25/RM3. Compared to OHSU7, however, the impact of synonymy on this query is limited.

A different situation occurs for CDS15 (Figure 6.11), where all SAFIR variants outperform BM25/RM3 in topic 22 (50% of polysemous words). The positive performances of all SAFIR variants – and in particular of SAFIR$_\mathrm{sp}$ – indicate that modeling both synonymy and polysemy is beneficial for this query.

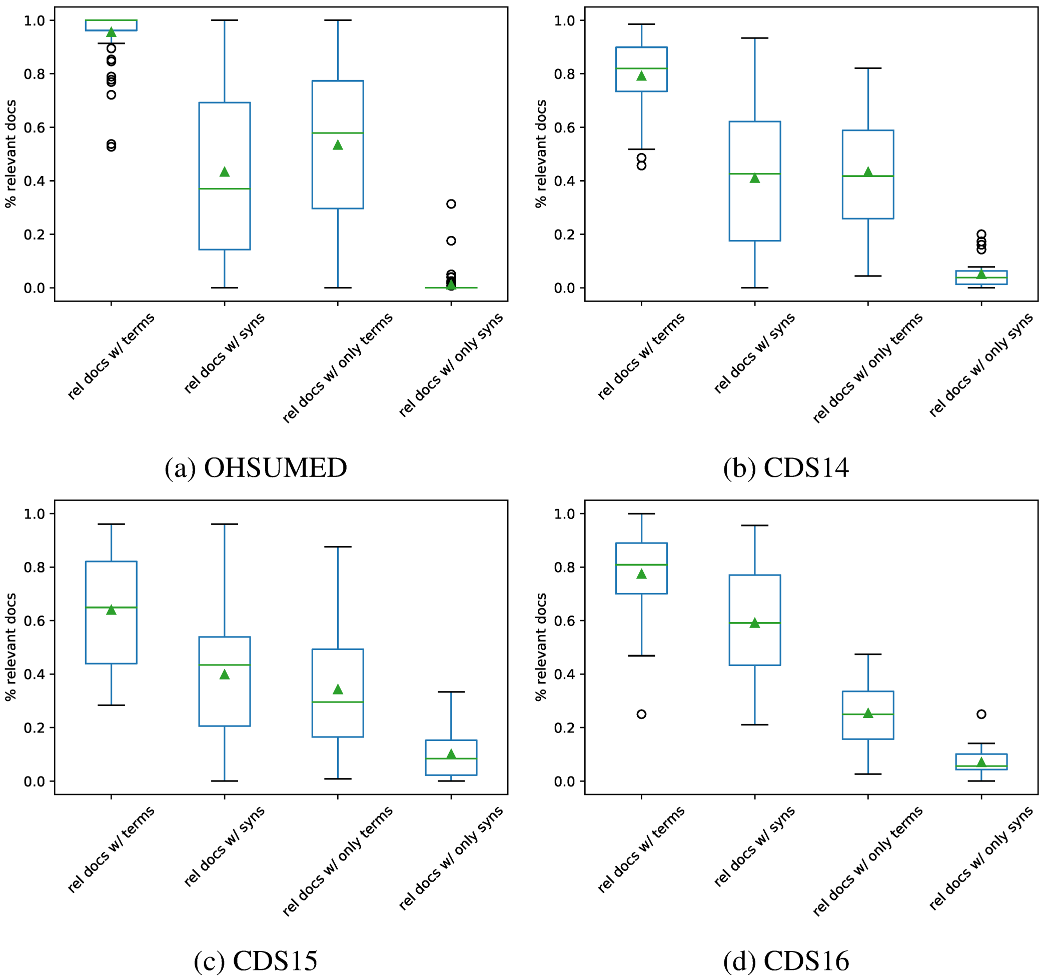
As for CDS16 (Figure 6.12), topic 20 – where 40% of the words are polysemous – shows similarities with topics OHSU7 (OHSUMED) and 26 (CDS14). Again, SAFIR$_\mathrm{p}$ and SAFIR$_\mathrm{sp}$ outperform BM25/RM3, while SAFIR$_\mathrm{s}$ does not. However, the differences between SAFIR$_\mathrm{p}$/SAFIR$_\mathrm{sp}$ and BM25/RM3 are small if compared to the other topics discussed. A possible reason could be the lower number of polysemous words within this query – which leads to a minor impact of polysemy on model performances.

Thus, the analysis confirms the effectiveness of modeling polysemy to answer semantically hard queries with a high degree of polysemy. SAFIR$_\mathrm{p}$ and SAFIR$_\mathrm{sp}$ effectively capture polysemy and, for highly polysemous queries, outperform BM25/RM3.

Synonymy Analysis

To identify the degree of synonymy in the considered collections, we account for (i) the proportion of relevant documents that contain at least one query term; (ii) the proportion of relevant documents that contain at least one synonym related to any query term; (iii) the proportion of relevant documents that contain only query terms; (iv) the proportion of relevant documents that contain only synonyms related to query terms.
Figure 6.13 shows the distribution of such proportions within each collection. Then, for each collection, we present one query where the integration of synonymy in the learning process produces effective results. Each query is selected to highlight this behavior and has the proportion of relevant documents containing synonyms close to or greater than the third quartile of the distribution generated from (ii). We report the results in Table 6.12 as a pairwise comparison between NVSM/SAFIR$_\mathrm{s}$ and SAFIR$_\mathrm{p}$/SAFIR$_\mathrm{sp}$. In this way, we emphasize the effectiveness of integrating synonymy by comparing pairs of models that rely on the same input data. As done for polysemy, we perform a second analysis where we compare SAFIR variants, NVSM, and BM25/RM3 on semantically hard queries. The objective is to understand whether the effectiveness of SAFIR$_\mathrm{s}$ and SAFIR$_\mathrm{sp}$ on queries with a large proportion of relevant documents containing only query synonyms holds against NVSM and BM25/RM3. For each collection, we consider the five queries that present the largest proportion of relevant documents containing only query synonyms (iv) and we present the results in Table 6.13.

The Degree of Synonymy. The distributions in Figure 6.13 provide two main insights. First, the proportion of relevant documents that contain only query synonyms is low for all queries in all collections. Therefore, modeling synonymy to retrieve relevant documents has a marginal impact on retrieval performances. Secondly, the proportion of relevant documents that contain at least one query synonym is, on average, lower than the proportion of relevant documents that contain at least one query term for all collections. Besides, the proportion of relevant documents that contain only query terms is, on average, close to the proportion of relevant documents that contain at least one query synonym for all collections but CDS16. In practice, this means that the impact of synonymy is also mitigated by the large number of relevant documents that contain (only) query terms. As a side note, the proportion of relevant documents that contain at least one query term has a median value of 1.0 in OHSUMED. This further explains the effectiveness of models relying on corpus-based features – and in particular of lexical models.



(a) OHSUMED

(b) CDS14

(c) CDS15

(d) CDS16

Fig. 6.13 Distribution of different proportions of relevant documents per topic. For each collection, the box-plots represent, from left to right, the distribution of the proportion of relevant documents per topic containing query terms, per topic containing synonyms related to query terms, per topic containing only query terms, and per topic containing only synonyms related to query terms.

Thus, the analysis explains the low – or even detrimental – impact of integrating synonymy and shows why SAFIR$_{s}$ and SAFIR$_{sp}$ present average performances close to or lower than those of NVSM and SAFIR$_{p}$, respectively. Nevertheless, we want to understand if modeling synonymy proves effective when the proportion of relevant documents that contain query synonyms is high. Our intuition is that for queries with a large number of relevant documents containing query synonyms, the effectiveness will be higher for SAFIR$_{s}$ and SAFIR$_{sp}$ than for NVSM and SAFIR$_{p}$, respectively.

Table 6.12 Pairwise comparison between SAFIR$_{s}$/NVSM and SAFIR$_{sp}$/SAFIR$_{p}$ on specific topics that present a large number of relevant documents containing query synonyms. For each measure, ↑/↓ means that the SAFIR variant integrating synonymy achieves higher/lower scores than its baseline.

OHSUMED - Topic OHSU22

	infNDCG	nDCG@1000	nDCG@100	nDCG@10	P@10	Recall@1000
NVSM	–	0.4267	0.2440	0.0526	0.1000	0.9600
SAFIR$_{s}$	–	0.4795$^{\uparrow}$	0.2906$^{\uparrow}$	0.1414$^{\uparrow}$	0.3000$^{\uparrow}$	1.0000$^{\uparrow}$
SAFIR$_{p}$	–	0.4177	0.2491	0.0435	0.1000	0.9600
SAFIR$_{sp}$	–	0.4457$^{\uparrow}$	0.2820$^{\uparrow}$	0.0473$^{\uparrow}$	0.1000	1.0000$^{\uparrow}$

CDS14 - Topic 24

	infNDCG	nDCG@1000	nDCG@100	nDCG@10	P@10	Recall@1000
NVSM	0.3118	0.5225	0.3300	0.2941	0.4000	0.7222
SAFIR$_{s}$	0.3797$^{\uparrow}$	0.5877$^{\uparrow}$	0.4018$^{\uparrow}$	0.4291$^{\uparrow}$	0.4000	0.7333$^{\uparrow}$
SAFIR$_{p}$	0.3281	0.5373	0.3472	0.3236	0.4000	0.7111
SAFIR$_{sp}$	0.3338$^{\uparrow}$	0.5631$^{\uparrow}$	0.3532$^{\uparrow}$	0.4019$^{\uparrow}$	0.5000$^{\uparrow}$	0.7222$^{\uparrow}$

CDS15 - Topic 19

	infNDCG	nDCG@1000	nDCG@100	nDCG@10	P@10	Recall@1000
NVSM	0.0075	0.1222	0.0092	0.0000	0.0000	0.2159
SAFIR$_{s}$	0.0171$^{\uparrow}$	0.1358$^{\uparrow}$	0.0209$^{\uparrow}$	0.0000	0.0000	0.2500$^{\uparrow}$
SAFIR$_{p}$	0.0088	0.0985	0.0108	0.0000	0.0000	0.1705
SAFIR$_{sp}$	0.0102$^{\uparrow}$	0.1243$^{\uparrow}$	0.0125$^{\uparrow}$	0.0000	0.0000	0.2273$^{\uparrow}$

CDS16 - Topic 26

	infNDCG	nDCG@1000	nDCG@100	nDCG@10	P@10	Recall@1000
NVSM	0.0899	0.2370	0.1058	0.0347	0.1000	0.3363
SAFIR$_{s}$	0.1288$^{\uparrow}$	0.2735$^{\uparrow}$	0.1359$^{\uparrow}$	0.1120$^{\uparrow}$	0.2000$^{\uparrow}$	0.3717$^{\uparrow}$
SAFIR$_{p}$	0.0385	0.1928	0.0453	0.0000	0.0000	0.3009
SAFIR$_{sp}$	0.0486$^{\uparrow}$	0.2084$^{\uparrow}$	0.0571$^{\uparrow}$	0.0000	0.0000	0.3186$^{\uparrow}$

The Impact of Modeling Synonymy. The results from Table 6.12 confirm this intuition and show the ability of SAFIR$_{\text{s}}$ and SAFIR$_{\text{sp}}$ to retrieve relevant documents that NVSM and SAFIR$_{\text{p}}$ fail to discover. In particular, SAFIR$_{\text{s}}$ and SAFIR$_{\text{sp}}$ achieve 100% Recall@1000 for topic OHSU22 (OHSUMED), thus retrieving all the relevant documents that neither NVSM nor SAFIR$_{\text{p}}$ discover. Furthermore, the results for nDCG measures show the ability of SAFIR$_{\text{s}}$ and SAFIR$_{\text{sp}}$ to effectively order relevant documents in the ranking list. For instance, SAFIR$_{\text{s}}$ achieves a 0.4291 of nDCG@10 in topic 24 (CDS14), whereas NVSM achieves 0.2941. Similarly, SAFIR$_{\text{sp}}$ achieves a 0.4019 of nDCG@10 that outperforms SAFIR$_{\text{p}}$. To a lesser extent, the results for topic 19 (CDS15) follow the same trend found in the topics analyzed for OHSUMED and CDS14. The main difference regards nDCG@10 and P@10 measures, where SAFIR$_{\text{s}}$ and SAFIR$_{\text{sp}}$ achieve the same performances of NVSM and SAFIR$_{\text{p}}$. In this case, SAFIR variants and NVSM fail to order relevant documents in the top positions of the ranking list. Finally, the results for topic 26 (CDS16) confirm the findings from the polysemy analysis on this query and highlight the effectiveness of integrating synonymy.

The Advantage of Modeling Synonymy. Given the outcomes of the previous analysis, we investigate whether the effectiveness of SAFIR$_{\text{s}}$ and SAFIR$_{\text{sp}}$ on semantically hard queries – i.e., queries with a large number of relevant documents containing only query synonyms – holds against NVSM and BM25/RM3. The results from Table 6.13 mark a clear distinction between OHSUMED, CDS16, and CDS14, CDS15. In OHSUMED and CDS16, BM25/RM3 achieves top performances for most measures. The only notable exception is Recall@1000, where SAFIR$_{\text{s}}$ and SAFIR$_{\text{sp}}$ outperform BM25/RM3 by a large margin. On the other hand, the results for CDS14 and CDS15 highlight the effectiveness of SAFIR$_{\text{s}}$ (CDS14) and SAFIR$_{\text{sp}}$ (CDS15) to answer semantically hard queries. In particular, SAFIR$_{\text{sp}}$ achieves top performances for all measures but infNDCG in CDS15.

If we analyze the proportion of relevant documents containing only query synonyms in the considered queries, we discover that the differences between OHSUMED, CDS16, and CDS14, CDS15 are related to such quantities. In both OHSUMED and CDS16, the number of relevant documents containing only query synonyms is less than 15% of the total number of relevant documents for three out of five queries. Conversely, CDS14 and CDS15 show proportions higher than 15% for most queries. In particular, all the five CDS15 queries present proportions greater than 20% – and close to 30% in three out of five cases.

Therefore, when the proportion of relevant documents containing only query synonyms is considerable, SAFIR$_{\text{s}}$ and SAFIR$_{\text{sp}}$ effectively capture synonymy and provide better results to semantically hard queries than NVSM and BM25/RM3 – which do not explicitly model

Table 6.13 Retrieval performances of SAFIR variants, NVSM, and BM25/RM3 on the five topics that present the largest number of relevant documents containing only query synonyms. For each measure, **bold** represents the model with the highest score.

	OHSUMED - Topics OHSU63, OHSU31, OHSU54, OHSU22, OHSU32					
	infNDCG	nDCG@1000	nDCG@100	nDCG@10	P@10	Recall@1000
BM25/RM3	–	**0.3926**	**0.2729**	**0.2480**	**0.3600**	0.6285
NVSM	–	0.3383	0.2248	0.1272	0.1800	0.5875
SAFIR_s	–	0.3892	0.2127	0.1860	0.2400	**0.6843**
SAFIR_p	–	0.3725	0.2438	0.1744	0.2200	0.6304
SAFIR_sp	–	0.3832	0.2457	0.1762	0.2200	0.6637

	CDS14 - Topics 16, 28, 13, 5, 24					
	infNDCG	nDCG@1000	nDCG@100	nDCG@10	P@10	Recall@1000
BM25/RM3	0.1241	0.2127	0.1280	**0.1482**	**0.2000**	0.29854
NVSM	0.1031	0.2236	0.1137	0.1175	0.1600	**0.3395**
SAFIR_s	**0.1268**	**0.2388**	**0.1436**	0.1434	0.1400	0.3313
SAFIR_p	0.1091	0.2184	0.1205	0.1165	0.1600	0.3124
SAFIR_sp	0.1149	0.2253	0.1323	0.1211	0.1600	0.3179

	CDS15 - Topics 7, 24, 16, 13, 11					
	infNDCG	nDCG@1000	nDCG@100	nDCG@10	P@10	Recall@1000
BM25/RM3	0.1080	0.1166	0.0976	0.1515	0.2000	0.1363
NVSM	0.1570	0.1785	0.1489	0.2357	0.3200	0.2154
SAFIR_s	0.1601	0.1822	0.1570	0.2575	0.3400	0.2202
SAFIR_p	**0.1794**	0.1979	0.1578	0.2612	0.3200	0.2295
SAFIR_sp	0.1768	**0.1986**	**0.1633**	**0.2878**	**0.3600**	**0.2348**

	CDS16 - Topics 22, 1, 5, 12, 13					
	infNDCG	nDCG@1000	nDCG@100	nDCG@10	P@10	Recall@1000
BM25/RM3	**0.1192**	**0.1701**	**0.1090**	**0.1443**	**0.2000**	0.2319
NVSM	0.0642	0.1469	0.0626	0.0712	0.1600	0.2492
SAFIR_s	0.0917	0.1503	0.0718	0.0861	0.1800	0.2270
SAFIR_p	0.0839	0.1628	0.0722	0.1021	0.1400	**0.2522**
SAFIR_sp	0.0900	0.1584	0.0683	0.1020	0.1400	0.2438

synonymy. However, compared to polysemy, the degree of synonymy is limited. Thus, the impact of modeling synonymy is marginal on average.

Take-home message. Modeling polysemy is effective and impacts the most when queries present a high degree of polysemy. On the other hand, the impact of synonymy on average performances is marginal – or even detrimental – due to the limited presence of relevant documents containing (only) query synonyms. Nevertheless, when we look at queries with a large number of relevant documents containing (only) query synonyms, SAFIR$_{\text{s}}$ and SAFIR$_{\text{sp}}$ capture synonymy and provide effective results.

6.7.2 The Effectiveness of Knowledge Resources for Document Retrieval

RQ2 How can external knowledge resources help to bridge the semantic gap between queries and documents?

When we compare knowledge-enhanced models with the corpus-driven baselines used as part of their learning process, we observe different trends. Regarding knowledge-enhanced baselines, we see that retrofitting models fail to enhance the baselines used as part of their learning process. rword2vec performs worse than word2vec for most measures in all collections. Similarly, rdoc2vec fails to improve on both its baselines and performs worse than doc2vec or cdoc2vec for most measures in all collections. The optimization function used by rdoc2vec retrofits document representations but leaves word and concept representations unchanged. Therefore, document and word/concept representations – that were jointly learned by doc2vec/cdoc2vec – misalign. This leads to a mismatch between retrofitted document representations and word/concept representations, which can explain the suboptimal performances achieved by rdoc2vec. On the other hand, the results show that cdoc2vec benefits from learning concepts rather than words for most measures in CDS collections. Conversely, cdoc2vec achieves significantly worse results than doc2vec in OHSUMED. The reason of this significant drop in performances can be attributed to how cdoc2vec builds query representations. In fact, cdoc2vec relies only on the concepts associated to the query terms to build query representations. Therefore, given the short length of OHSUMED queries, this building process leads to noneffective representations.

As for SAFIR, we see that all the variants outperform NVSM for most measures in all collections. Depending on the measure and collection considered, different SAFIR variants achieve the best results. Interestingly, the results for nDCG@10 show that all SAFIR variants order relevant documents in top positions better than NVSM. This highlights the effectiveness, for precision-oriented measures, of integrating external knowledge while optimizing word, concept, and document representations for retrieval. Of all the SAFIR variants, NVSM gets closer to SAFIR$_s$ performances. In particular, SAFIR$_s$ performs worse than NVSM for nDCG@1000 and Recall@1000 both in CDS14 and CDS16. As seen in the synonymy analysis, SAFIR$_s$ performance is impacted by the limited presence of relevant documents containing (only) query synonyms. However, the fact that SAFIR$_s$ outperforms NVSM for infNDCG in all CDS collections suggests that, with a larger number of relevance judgments, SAFIR$_s$ could achieve higher nDCG values than NVSM. Indeed, we recall that infNDCG provides a better estimate of the real value of nDCG in case of incomplete relevance judgments [249].

Compared to the other knowledge-enhanced models, we emphasize the effectiveness of SAFIR for Recall@1000 and nDCG measures. Recall@1000 shows the ability of SAFIR variants to retrieve relevant documents while nDCG measures show how well these documents are ranked at different cutoff levels. Therefore, we perform the following analyses to further evaluate the differences between SAFIR and the considered baselines.

Knowledge-Enhanced Relevance Analysis

We analyze the number of relevant documents retrieved by SAFIR variants and knowledge-enhanced baselines. For each topic of CDS collections, we compare the number of exclusive relevant documents that only SAFIR retrieves with respect to the union of the relevant documents retrieved by all the knowledge-enhanced baselines. This means that we compare SAFIR against a “fictitious and boosted” model that considers all the relevant documents retrieved by the knowledge-enhanced baselines. We adopt this solution instead of comparing SAFIR individually with each knowledge-enhanced baseline to save space and compare SAFIR with a highly challenging baseline. Figure 6.14 reports the per-topic results of this analysis.

The analysis shows how exclusive SAFIR is in retrieving relevant documents that none of the knowledge-enhanced baselines retrieve. SAFIR retrieves more exclusive relevant documents than all the other knowledge-enhanced models together. In particular, SAFIR variants retrieve more exclusive relevant documents for almost all the topics of CDS14 and CDS16 collections. The exclusiveness of SAFIR is less evident in CDS15, as the impact of rword2vec in the “fictitious” model reduces the gap with SAFIR. If we analyze the per-topic behavior of each SAFIR variant, we observe that, with the due differences, all variants have a similar trend in terms of exclusive relevant documents retrieved. This suggests that SAFIR prioritizes text matching when learning representations and relies on polysemy and synonymy to refine such representations towards one, or both, features.

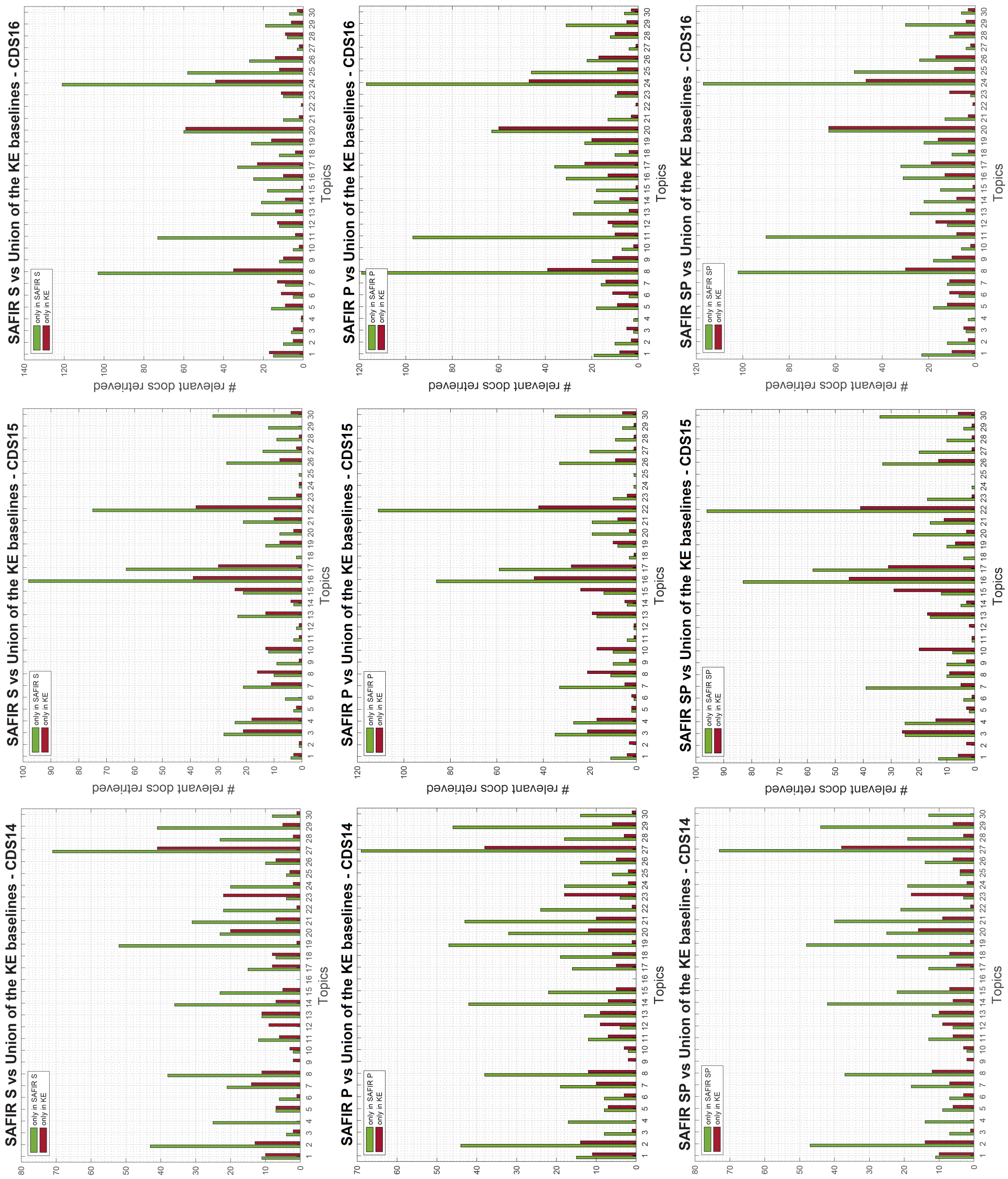


Fig. 6.14 Per-topic analysis of the number of relevant documents retrieved by SAFIR variants and by the union of the knowledge-enhanced baselines. For each topic, the green (light) bar represents the number of relevant documents that only SAFIR retrieves and the red (dark) bar represents the union of the number of relevant documents that the knowledge-enhanced baselines retrieve.

We present three examples – one for each SAFIR variant – where we qualitatively analyze the impact of knowledge resources in modeling synonymy, polysemy, or both. Each example represents a query where a particular SAFIR variant retrieves the highest number of exclusive relevant documents compared to the other variants and the fictitious model. We discuss the aspects of each example related to synonymy and/or polysemy and we provide insights on why a particular SAFIR variant performs best. Then, we present a fourth example where SAFIR is outperformed by the fictitious model, which retrieves more exclusive relevant documents.

Example 6.7.1. CDS14 Topic 4: "*2-year-old boy with fever and irritability for 5 days. Physical exam findings include conjunctivitis, strawberry tongue, and desquamation of the fingers and toes. Lab results include low albumin, elevated white blood cell count and C-reactive protein, and urine leukocytes. Echo shows moderate dilation of the coronary arteries.*"

- $\{\text{SAFIR}_\text{s}\} \setminus \{\text{knowledge-enhanced baselines (union)}\}$: 25
- $\{\text{knowledge-enhanced baselines (union)}\} \setminus \{\text{SAFIR}_\text{s}\}$: 0

For topic 4 of CDS14, SAFIR_s retrieves twenty-five documents that the fictitious model does not retrieve. Conversely, the fictitious model does not retrieve any relevant documents that SAFIR_s does not retrieve. For this query, an interesting example is provided by document `3152734`. Document `3152734` describes common associated symptoms (e.g., strawberry tongue) and their clinical significance in children affected with the Kawaski disease. The document contains words like "children" and "febrile", which convey the same meaning of query words "boy" and "fever". Therefore, by modeling synonymy, SAFIR_s reduces the semantic gap between the query and this (relevant) document and improves retrieval. This document is not retrieved by SAFIR_p, which does not model synonymy, and neither by lexical models, since query words are not contained within it.

Example 6.7.2. CDS15 Topic 22: "*A 65-year-old male complains of productive cough with tinges of blood. Chest X-ray reveals a round opaque mass within a cavity in his lung. Culture of the sputum revealed fungal elements.*"

- $\{\text{SAFIR}_\text{p}\} \setminus \{\text{knowledge-enhanced baselines (union)}\}$: 111
- $\{\text{knowledge-enhanced baselines (union)}\} \setminus \{\text{SAFIR}_\text{p}\}$: 42

For topic 22 of CDS15, SAFIR_p retrieves 111 documents that the fictitious model does not retrieve. On the other hand, the fictitious model retrieves forty-two relevant documents that SAFIR$_\text{p}$ does not. Among the unique relevant documents retrieved by SAFIR$_\text{p}$, document 3014676 presents interesting aspects. Document 3014676 describes treatments for the allergic bronchopulmonary aspergillosis. The disease derives from the Aspergillus, a soil-dwelling fungus known to cause significant pulmonary infection in immunocompromised patients. The document presents various acronyms and morphosyntactic variants. In particular, the acronym "ABPA" – which stands for "Allergic Bronchopulmonary Aspergillosis" – can be especially ambiguous for an automatic system. Indeed, within UMLS the acronym "ABPA" can be associated to five different meanings (CUIs) like: "Aspergillosis, Allergic Bronchopulmonary" (C0004031), "FLNC gene" (C1414637), and "AbpA protein, Streptococcus gordonii" (C1308582). Therefore, to relate such word to discriminative words within the query (e.g., the query words "lung" and "fungal") it is important to disambiguate its meaning. By modeling polysemy, SAFIR$_\text{p}$ removes this ambiguity in document and query words and improves retrieval. It is worth mentioning that this document is not retrieved by SAFIR$_\text{s}$, which does not model polysemy, and neither by lexical models.

Example 6.7.3. CDS16 Topic 14: "*A 52 year-old woman with history of COPD and breast cancer who presents with SOB, hypoxia, cough, fevers and sore throat for several weeks.*"

- $\{\text{SAFIR}_\text{sp}\} \setminus \{\text{knowledge-enhanced baselines (union)}\}$: 22
- $\{\text{knowledge-enhanced baselines (union)}\} \setminus \{\text{SAFIR}_\text{sp}\}$: 8

For topic 14 of CDS16, SAFIR$_\text{sp}$ retrieves twenty-two documents that the fictitious model does not retrieve. Instead, the fictitious model retrieves eight relevant documents that SAFIR$_\text{sp}$ does not find. The query presents two interesting acronyms: COPD and SOB. The former stands for chronic obstructive pulmonary disease, whereas the latter for shortness of breath. COPD is a type of obstructive lung disease characterized by long-term breathing problems and poor airflow. In COPD, shortness of breath is a common respiratory symptom. Therefore, both acronyms need to be correctly disambiguated to retrieve relevant documents associated with them. We focus on document 3266210, which describes a clinical trial for the treatment of COPD. In particular, document 3266210 contains the word "dyspnoea" – which is a synonym of SOB. Thus, by modeling both synonymy and polysemy together, SAFIR$_\text{sp}$ encodes semantic features required to effectively retrieve this document. Interestingly, SAFIR$_\text{sp}$ is the only variant retrieving this document.

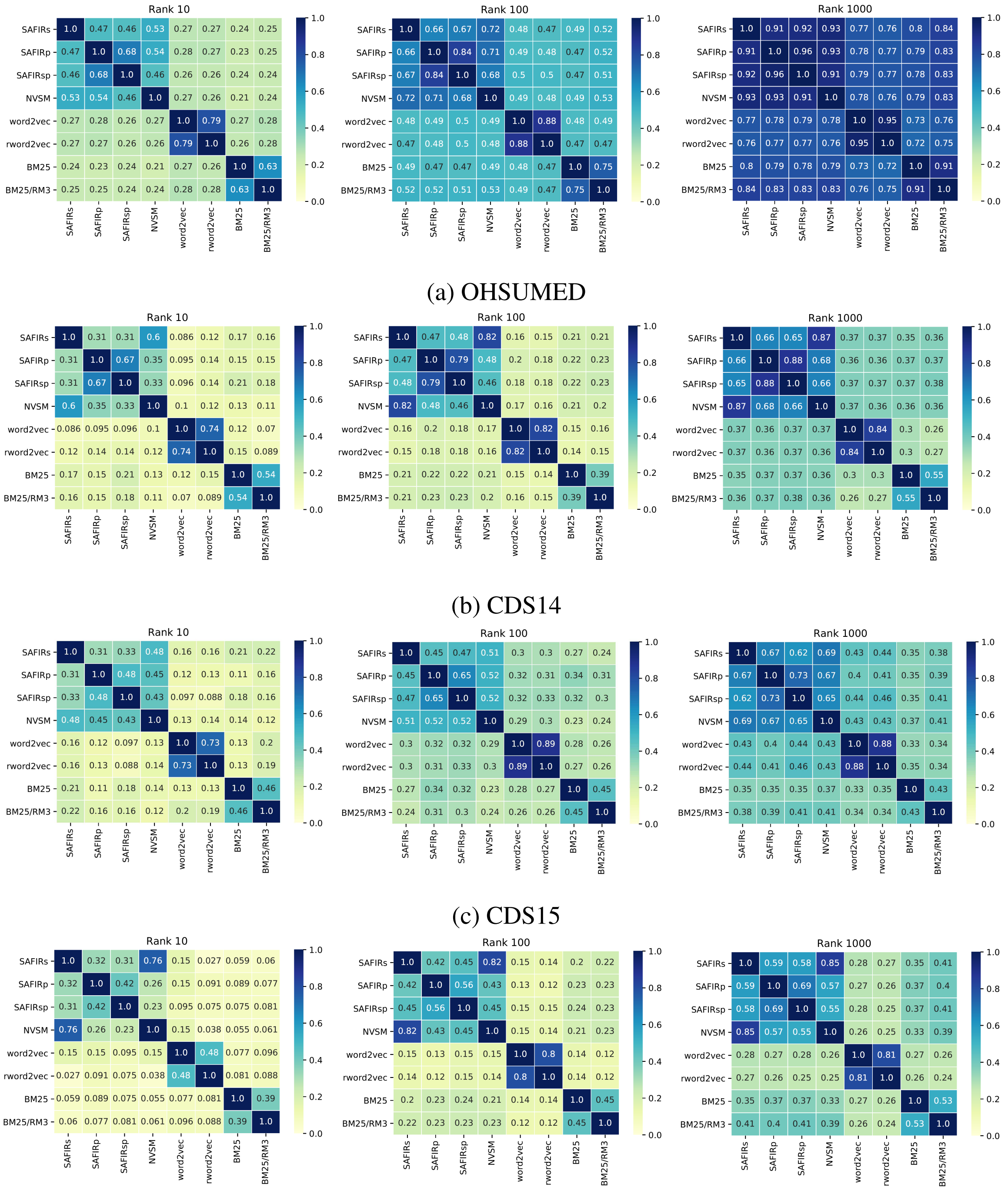
Example 6.7.4. CDS14 Topic 23: "*63-year-old heavy smoker with productive cough, shortness of breath, tachypnea, and oxygen requirement. Chest x-ray shows hyperinflation with no consolidation.*"

For topic 23 of CDS14, none of the SAFIR variants retrieve more than four documents that the fictitious model does not retrieve. In particular, SAFIR_s and SAFIR_p retrieve four documents that the fictitious model does not find, whereas SAFIR_sp only three. On the other hand, the fictitious model retrieves twenty-two documents that SAFIR_s does not retrieve and eighteen documents that neither SAFIR_p nor SAFIR_sp retrieve. It is worth mentioning that the knowledge-enhanced baseline that impacts the most within the fictitious model is rword2vec. In fact, rword2vec retrieves nineteen of the twenty-two documents that SAFIR_s does not find and sixteen of the eighteen documents that neither SAFIR_p nor SAFIR_sp discover.

Relevance Similarity Analysis

We evaluate to what extent SAFIR variants and the considered baselines retrieve different relevant documents. For each collection and pair of models, we compute the mean Jaccard index between the sets of relevant documents retrieved at different cutoffs. Given a pair of models, we compute the per-topic Jaccard index as the cardinality of the intersection divided by the cardinality of the union of the sets of relevant documents retrieved by the two considered models at a given cutoff. Then, the mean Jaccard index takes the average of the per-topic indices computed at the corresponding cutoff. When computing the mean Jaccard index, we do not count topics where none of the two considered models retrieve relevant documents (i.e., missing values). We use the same cutoff values used for nDCG measures, that is 10, 100, and 1000. In particular, we evaluate the degree of similarity between models that exhibit similar average performances. We want to understand to what extent these models retrieve different relevant documents. We do not report Jaccard index values for QLM and doc2vec models to save space and ease visualization. The performances of QLM are always comparable or lower than those of BM25, whereas doc2vec models never belong to the top statistical group (†).

Figure 6.15 shows the heatmaps of the mean Jaccard indices between the sets of relevant documents retrieved by each pair of models across topics at cutoffs 10, 100, and 1000, respectively, for each collection. The heatmaps highlight three clusters of models with higher similarity scores. The first cluster is composed of SAFIR variants and NVSM, the second of word2vec and rword2vec, whereas the third one comprises BM25 and BM25/RM3. Within the first cluster, the NVSM/SAFIR_s and SAFIR_p/SAFIR_sp pairs show higher scores due to the inherent similarity between the models. Nevertheless, we observe that NVSM/SAFIR_s and SAFIR_p/SAFIR_sp pairs never exceed a similarity score of 0.70 at cutoff 10. The only exception is in CDS16, where the NVSM/SAFIR_s pair shows a similarity score of 0.76. Therefore, all these models retrieve a significant number of different relevant documents in top positions of the ranking list. This behavior is even more pronounced when we consider



(a) OHSUMED

(b) CDS14

(c) CDS15

(d) CDS16

Fig. 6.15 Heatmaps of the mean Jaccard indices between the sets of relevant documents retrieved by each pair of models across topics at cutoffs 10, 100, and 1000, respectively, for each collection.

the similarity scores between NVSM and either SAFIR$_\mathrm{p}$ or SAFIR$_\mathrm{sp}$. In fact, the scores for the NVSM/SAFIR$_\mathrm{p}$ and NVSM/SAFIR$_\mathrm{sp}$ pairs keep low for all cutoffs in CDS collections – never exceeding values of 0.50, 0.55, and 0.70 at cutoffs 10, 100, and 1000, respectively. Within the second cluster, word2vec and rword2vec consistently present the same level of similarity for all cutoffs in all collections. The only exception is in CDS16, where they show a similarity score of 0.48 at rank 10. Within the third cluster, BM25 and BM25/RM3 show similarity scores lower than 0.60 in all CDS collections – regardless of the cutoff. This reflects the impact of expanding the original query with RM3, which enables BM25 to discover more relevant documents compared to the first round of retrieval.

Outside the clusters, the low similarity scores in CDS collections indicate that all the models retrieve different relevant documents regardless of the cutoff. Conversely, all the considered pairs present high similarity scores at cutoffs 100 and 1000 in OHSUMED. We attribute this behavior to two main reasons: (i) the high proportion of relevant documents that contain at least one query term in OHSUMED (see Figure 6.13), which favors models relying on corpus-based features; (ii) the small size of corpus and vocabulary in OHSUMED (see Table 6.7), which reduces the amount of polysemous and synonymous words within the collection.

Thus, the results show how different models are in terms of relevant documents retrieved. In particular, SAFIR variants and NVSM significantly differ in the relevant documents retrieved at cutoff 10 in CDS collections. SAFIR$_\mathrm{p}$ and SAFIR$_\mathrm{sp}$ keep this behavior also at cutoffs 100 and 1000, whereas SAFIR$_\mathrm{s}$ becomes similar to NVSM. This means that, even though SAFIR$_\mathrm{p}$, SAFIR$_\mathrm{sp}$, and NVSM present similar average performances at cutoffs 100 and 1000 (see Table 6.11), they achieve such performances by retrieving different relevant documents. On the other hand, the low similarity between SAFIR variants and BM25/RM3 – in terms of relevant documents retrieved – highlights the difference between semantic models and PRF based methods in addressing the semantic gap. This suggests that SAFIR and RM3 can be used as complementary approaches to address the semantic gap. Therefore, in the next section, we investigate the effectiveness of SAFIR variants in retrieving feedback documents for PRF based methods.

Take-home message. The integration of knowledge resources into the learning process of neural IR models is effective and helps to bridge the semantic gap between queries and documents. The learned representations encode text matching signals, necessary for IR tasks, and linguistic features to retrieve relevant documents that are most affected by the semantic gap. In particular, integrating external knowledge helps to boost the results at the top positions of the ranking list.

6.8 Query Expansion: Experimental Results

We present the experimental results for query expansion and we discuss them based on the research questions. Table 6.14 shows the performances for query expansion. We do not report the results of RM3 with QLM for the sake of simplicity. Indeed, the performances with QLM were always comparable or lower than those obtained using BM25. Also, we do not consider doc2vec-based models because of their poor performances.

6.8.1 The Impact of Polysemy and Synonymy on Query Expansion

RQ1 Which feature between synonymy and polysemy can be exploited to reduce the semantic gap and improve retrieval?

We see from Table 6.14 that for most measures there is no statistical difference among all RM3-enhanced models. In particular, RM3-enhanced models do not present statistical differences for P@10, nDCG@10, and infNDCG in all collections. Besides, in cases where there is statistical significance, the only RM3-enhanced models that do not belong to the top group are those using word2vec and rword2vec for the first round of retrieval. Nevertheless, RM3-enhanced models based on SAFIR variants achieve the best results for most measures in CDS collections. The only exceptions are in CDS16, where the RM3-enhanced model using BM25 for both rounds of retrieval achieve better performances in Recall@1000, nDCG@100, and P@10.

Among SAFIR variants, SAFIR$_{\text{sp}}$ provides expansion terms that allow BM25 to achieve the best scores for most measures in CDS collections. This is an interesting result as it shows that modeling both synonymy and polysemy is effective to retrieve feedback documents from which expansion terms are extracted. In other words, SAFIR$_{\text{sp}}$ helps BM25 to bridge the semantic gap more effectively than the other models for CDS collections. Even when different RM3-enhanced models achieve better performances, like SAFIR$_{\text{s}}$/RM3 and SAFIR$_{\text{p}}$/RM3 in CDS14 or BM25/RM3 in CDS16, the improvements over SAFIR$_{\text{sp}}$/RM3 are small in most cases.

Given that SAFIR$_{\text{p}}$ outperforms SAFIR$_{\text{sp}}$ for nDCG@10 in all CDS collections (see Table 6.11), we provide the following explanation to motivate the higher effectiveness of SAFIR$_{\text{sp}}$/RM3 compared to SAFIR$_{\text{p}}$/RM3. First of all, the differences between SAFIR$_{\text{sp}}$ and SAFIR$_{\text{p}}$ for nDCG@10 are small. This means that SAFIR$_{\text{sp}}$ and SAFIR$_{\text{p}}$ have similar effectiveness in retrieving and ordering relevant documents in top positions of the ranking list. On the other hand, SAFIR$_{\text{sp}}$ and SAFIR$_{\text{p}}$ significantly differ in the relevant documents retrieved at cutoff 10 (see Figure 6.15). In particular, the similarity score between the two

Table 6.14 RM3-enhanced models performances. RM3-enhanced models are grouped by the type of the model used in the first round of retrieval: Bag-of-Words (BoW), Corpus-Driven (CD), Knowledge-Enhanced (KE), and SAFIR. BM25 is always used for the second round of retrieval. The scores in parentheses represent the scores achieved by the model used in the first round of retrieval. **Bold** values represent the highest scores among RM3-enhanced models in each collection. † represents the models belonging to the statistical top group for the given collection with $\alpha \leq 0.05$.

		infNDCG				nDCG@1000			
		CDS14	CDS15	CDS16	OHSUMED	CDS14	CDS15	CDS16	OHSUMED
BoW	BM25/RM3	0.1384 (0.1064)	0.1578 (0.1276)	0.1688 (0.1399)	– (–)	0.2316† (0.1838)	0.2183 (0.1579)	0.2068† (0.1643)	0.6253 (0.5875)
CD	word2vec/RM3	0.1157 (0.0954)	0.1403 (0.1159)	0.1292 (0.0928)	– (–)	0.1895 (0.1548)	0.2061 (0.1634)	0.1492 (0.1054)	0.6507 (0.5902)
	NVSM/RM3	0.1673 (0.1576)	0.1453 (0.1449)	0.1425 (0.1475)	– (–)	0.2724† (0.2649)	0.2081 (0.2213)	0.1941† (0.1818)	0.6511 (0.5977)
KE	rword2vec/RM3	0.1100 (0.0896)	0.1383 (0.1142)	0.1314 (0.0790)	– (–)	0.1836 (0.1501)	0.2063 (0.1589)	0.1531 (0.0980)	**0.6539** (0.5852)
SAFIR	SAFIR$_s$/RM3	0.1680 (0.1602)	0.1582 (0.1498)	0.1490 (0.1546)	– (–)	0.2774† (0.2546)	0.2236 (0.2240)	0.1942† (0.1783)	0.6509 (0.6046)
	SAFIR$_p$/RM3	**0.1899** (0.1608)	0.1660 (0.1516)	0.1463 (0.1523)	– (–)	**0.2979**† (0.2723)	0.2276 (0.2247)	0.1975† (0.1858)	0.6477 (0.6106)
	SAFIR$_{sp}$/RM3	0.1898 (0.1566)	**0.1756** (0.1515)	**0.1726** (0.1599)	– (–)	0.2948† (0.2651)	**0.2410** (0.2266)	**0.2092**† (0.1849)	0.6470 (0.6144)

		nDCG@100				nDCG@10			
		CDS14	CDS15	CDS16	OHSUMED	CDS14	CDS15	CDS16	OHSUMED
BoW	BM25/RM3	0.1338 (0.1098)	0.1522 (0.1233)	**0.1298**† (0.1078)	0.4746 (0.4392)	0.1645 (0.1530)	0.1986 (0.2166)	0.1518 (0.1606)	0.4618 (0.4429)
CD	word2vec/RM3	0.1025 (0.0821)	0.1302 (0.1064)	0.0893 (0.0619)	0.5010 (0.4461)	0.1346 (0.1028)	0.1705 (0.1435)	0.1121 (0.0977)	0.4985 (0.4754)
	NVSM/RM3	0.1548 (0.1362)	0.1374 (0.1385)	0.1076† (0.1077)	0.4956 (0.4181)	0.2060 (0.1694)	0.1702 (0.1664)	0.1296 (0.1324)	**0.4989** (0.3873)
KE	rword2vec/RM3	0.0970 (0.0774)	0.1308 (0.1032)	0.0922 (0.0590)	**0.5069** (0.4421)	0.1180 (0.0967)	0.1742 (0.1410)	0.1175 (0.0930)	0.4977 (0.4709)
SAFIR	SAFIR$_s$/RM3	0.1585 (0.1385)	0.1521 (0.1411)	0.1106† (0.1071)	0.4962 (0.4216)	0.2229 (0.1729)	0.2033 (0.1818)	0.1249 (0.1374)	0.4902 (0.4121)
	SAFIR$_p$/RM3	0.1724 (0.1435)	0.1594 (0.1395)	0.1102† (0.1113)	0.4941 (0.4361)	0.2185 (0.1931)	0.2046 (0.2053)	0.1225 (0.1519)	0.4911 (0.4267)
	SAFIR$_{sp}$/RM3	**0.1762** (0.1401)	**0.1696** (0.1403)	0.1219† (0.1098)	0.4948 (0.4397)	**0.2253** (0.1898)	**0.2407** (0.1926)	**0.1572** (0.1475)	0.4856 (0.4380)

		P@10				Recall@1000			
		CDS14	CDS15	CDS16	OHSUMED	CDS14	CDS15	CDS16	OHSUMED
BoW	BM25/RM3	0.1833 (0.1667)	0.2433 (0.2600)	**0.2067** (0.2167)	0.5413 (0.5016)	0.3151† (0.2503)	0.2884 (0.1826)	**0.3059**† (0.2286)	0.8431 (0.7973)
CD	word2vec/RM3	0.1667 (0.1133)	0.2233 (0.1900)	0.1300 (0.1167)	0.5651 (0.5048)	0.2770 (0.2200)	0.2795 (0.2194)	0.2185 (0.1515)	0.8644 (0.7778)
	NVSM/RM3	0.2400 (0.2033)	0.2333 (0.2333)	0.1767 (0.1600)	0.5603 (0.4333)	0.3760† (0.3833)	0.2882 (0.3093)	0.2822† (0.2617)	0.8669 (0.8584)
KE	rword2vec/RM3	0.1500 (0.1267)	0.2400 (0.1967)	0.1433 (0.1133)	**0.5714** (0.5048)	0.2712 (0.2221)	0.2816 (0.2151)	0.2239 (0.1414)	0.8680 (0.7672)
SAFIR	SAFIR$_s$/RM3	**0.2600** (0.1967)	0.2667 (0.2267)	0.1667 (0.1733)	0.5587 (0.4619)	0.3853† (0.3607)	0.2970 (0.3134)	0.2814† (0.2545)	**0.8755** (0.8582)
	SAFIR$_p$/RM3	0.2433 (0.2333)	0.2600 (0.2633)	0.1600 (0.1700)	0.5698 (0.4762)	**0.4100**† (0.3846)	0.3006 (0.3098)	0.2866† (0.2782)	0.8687 (0.8548)
	SAFIR$_{sp}$/RM3	0.2433 (0.2200)	**0.3067** (0.2467)	0.1933 (0.1633)	0.5571 (0.4794)	0.3991† (0.3733)	**0.3134** (0.3110)	0.3047† (0.2747)	0.8708 (0.8520)

models is lower than 0.50 in both CDS15 and CDS16. Thus, our intuition is that SAFIR$_{\text{sp}}$ – by modeling both polysemy and synonymy – retrieves feedback documents that provide better expansion terms than those retrieved by SAFIR$_{\text{p}}$. To support this intuition, let us consider topic 25 from CDS16. For this query, the difference between SAFIR$_{\text{sp}}$ and SAFIR$_{\text{p}}$ in terms of nDCG@10 is close to zero (see Figure 6.8), whereas SAFIR$_{\text{sp}}$/RM3 outperforms SAFIR$_{\text{p}}$/RM3 by a large margin (> 0.60) for infNDCG, nDCG@10, and P@10.

CDS16 Topic 25: *"An elderly female with history of atrial fibrillation, Chronic Obstructive Pulmonary Disease, hypertension, hyperlipidemia and previous repair of atrial septum defect, presenting with shortness of breath and atrial fibrillation resistant to medication."*

The query describes a woman presenting shortness of breath and atrial fibrillation, with history of arrhythmia and other correlated diseases. In this case, SAFIR$_{\text{sp}}$ and SAFIR$_{\text{p}}$ provide six exclusive expansion terms:

SAFIR$_{\text{sp}}$: amiodarone, cardiac, dronedarone, metaprolol, procedure, rhythm

SAFIR$_{\text{p}}$: atrium, cha, chads, flutter, hypertensive, permanent

The expansion terms from SAFIR$_{\text{sp}}$ show a higher diversity than those from SAFIR$_{\text{p}}$. Three of these terms, namely "amiodarone", "dronedarone", and "metaprolol", refer to drugs or medications used to treat (and prevent) a number of types of arrhythmia – including atrial fibrillation. The other terms are highly correlated ("rhythm") and help to contextualize the anatomical region of interest ("cardiac"). On the other hand, three of the six exclusive terms provided by SAFIR$_{\text{p}}$ are terminological variants of the query terms – that is, "atrium", "flutter", and "hypertensive". As for the other terms, both "cha" and "chads" refer to clinical prediction rules used to estimate the risk of stroke in patients with atrial fibrillation. Thus, although both SAFIR variants provide highly related expansion terms, those obtained using SAFIR$_{\text{sp}}$ have more variety and help BM25 to bridge the semantic gap more effectively.

A different situation occurs in OHSUMED, where all the RM3-enhanced models show similar results. Depending on the measure, different RM3-enhanced models achieve the best results. In particular, rword2vec/RM3 provides the highest scores for nDCG@1000, nDCG@100, and P@10. In this case, SAFIR$_{\text{s}}$ and SAFIR$_{\text{p}}$ provide better expansion terms than SAFIR$_{\text{sp}}$ as both SAFIR$_{\text{s}}$/RM3 and SAFIR$_{\text{p}}$/RM3 achieve higher scores than SAFIR$_{\text{sp}}$/RM3 for most measures. Nevertheless, the only RM3-enhanced model relying on SAFIR that achieves top performances is SAFIR$_{\text{s}}$/RM3 for Recall@1000. Thus, the results strengthen our hypothesis that OHSUMED favors models relying on corpus-based features, such as rword2vec.

Take-home message. The effectiveness of SAFIR$_{\text{sp}}$ to provide expansion terms that help BM25 to fill the semantic gap in CDS collections shows the importance of modeling both synonymy and polysemy.

6.8.2 The Effectiveness of Knowledge Resources for Query Expansion

RQ2 How can external knowledge resources help to bridge the semantic gap between queries and documents?

We see that SAFIR provides better expansion terms than BM25. In fact, RM3-enhanced models relying on SAFIR variants achieve higher results than BM25/RM3 for most measures in all collections. Furthermore, all the SAFIR-based RM3-enhanced models outperform NVSM/RM3 for most measures in CDS collections. Thus, the effectiveness of SAFIR variants for nDCG@10 (see Table 6.11), along with their exclusiveness in terms of relevant documents retrieved (see Figure 6.15), make them more suitable than BM25 or NVSM to perform the first round of retrieval in RM3-enhanced models.

Regarding rword2vec, we see that rword2vec/RM3 achieves performances higher than word2vec/RM3 for many measures in all collections. However, the differences between rword2vec/RM3 and word2vec/RM3 are less prominent than those between SAFIR-based RM3-enhanced models and NVSM/RM3. Nevertheless, we advocate that knowledge-enhanced models provide better expansion terms than corpus-driven ones.

Take-home message. Knowledge-enhanced models grasp different signals than lexical or corpus-driven models and retrieve documents in top positions that are effective in providing expansion terms for PRF models.

6.9 Chapter Outcomes and Lessons Learned

In this chapter, we have investigated how to integrate external knowledge into the learning process of neural models to reduce the effect of the semantic gap between queries and documents. We have focused on medical literature and we have considered two linguistic features related to the semantic gap: synonymy and polysemy.

First, we have performed a reproducibility study of the works by Liu et al. [147] and Nguyen et al. [169]. The knowledge-enhanced word embeddings used, and proposed, by Liu et al. model synonymy through a knowledge-based regularization during/after the training of word representations. On the other hand, the knowledge-enhanced document embeddings proposed by Nguyen et al. [169] model both synonymy and polysemy through a retrofitting approach that exploits document representations learned using words and concepts. The reproducibility study highlighted some limitations in the way knowledge-enhanced neural language models are applied to IR, as well as in their effectiveness to address the semantic gap between queries and documents. In particular, the outcomes of this study emphasized the inability of neural language models to effectively encode relevant features for IR, along with the need for knowledge-enhanced neural IR models capable of providing effective performances at the early stages of the IR pipeline – where the integration of external knowledge can express its full potential.

Then, motivated by the outcomes of the reproducibility study, we have introduced the Semantic-Aware Neural Framework for IR (SAFIR), an unsupervised knowledge-enhanced neural framework for IR. SAFIR jointly learns word, concept, and document representations from scratch. The learned representations are optimized for IR and encode polysemy and/or synonymy with the aim of addressing the semantic gap between queries and documents. Regarding polysemy, SAFIR contextualizes word meanings by combining word and concept representations in the learning process. Thus, word meaning representations are created on-the-fly by combining word and concept representations. This compositional process avoids the creation of a representation for each word meaning. Then, word and concept representations are optimized to minimize the distance between the word meanings and the documents in the vector space. At the same time, SAFIR models synonymy via multi-task learning. Word representations are shared between text matching and word similarity tasks. For the word similarity task, SAFIR minimizes the distance between word representations for words presenting synonymy relations within an external knowledge resource.

For evaluation, we considered three variants of SAFIR: SAFIR$_{\text{sp}}$, which integrates both synonymy and polysemy; SAFIR$_{\text{s}}$ which integrates synonymy but not polysemy; and SAFIR$_{\text{p}}$ which integrates polysemy but not synonymy. We compared SAFIR variants with knowledge-enhanced and corpus-driven neural models on medical literature retrieval considering two strategies: document retrieval and query expansion. The experimental results we obtained led to the following conclusions:

RQ1 Which feature between synonymy and polysemy can be exploited to reduce the semantic gap and improve retrieval?

Document Retrieval: modeling polysemy is effective and impacts the most when queries present a high degree of polysemy. On the other hand, the impact of synonymy on average performances is marginal – or even detrimental – due to the limited presence of relevant documents containing (only) query synonyms. Nevertheless, when we look at queries with a large number of relevant documents containing (only) query synonyms, SAFIR$_s$ and SAFIR$_{sp}$ capture synonymy and provide effective results.

Query Expansion: the effectiveness of SAFIR$_{sp}$ to provide expansion terms that help BM25 to fill the semantic gap in CDS collections shows the importance of modeling both synonymy and polysemy.

RQ2 How can external knowledge resources help to bridge the semantic gap between queries and documents?

Document Retrieval: the integration of knowledge resources into the learning process of neural IR models is effective and helps to bridge the semantic gap between queries and documents. The learned representations encode text matching signals, necessary for IR tasks, and linguistic features to retrieve relevant documents that are most affected by the semantic gap. In particular, integrating external knowledge helps to boost the results at the top positions of the ranking list.

Query Expansion: knowledge-enhanced neural models grasp different signals than lexical models or corpus-driven neural models and retrieve documents in top positions that are effective in providing expansion terms for Pseudo Relevance Feedback (PRF) models.

The evaluation showed that SAFIR retrieves more exclusive relevant documents than knowledge-enhanced neural language models for most queries in all collections. Furthermore, the effectiveness of SAFIR for precision-oriented measures, along with its exclusiveness in terms of relevant documents retrieved, makes it suitable for PRF based methods. Therefore, our evaluation suggests that unsupervised knowledge-enhanced semantic models should be used at the early stages of the IR pipeline rather than in re-ranking scenarios – where interaction-based re-ranking models could easily outperform them (e.g., see DRMM [95] performance in Chapter 4). In this way, the different signals that knowledge-enhanced semantic models provide can be used by multi-stage IR systems to obtain a richer pool of relevant documents, thus leading to better answers for semantically hard queries.

Chapter 7

Conclusion and Future Work

The leitmotiv of this thesis is the investigation of the semantic gap and how it can be addressed to improve retrieval performance. The semantic gap represents the mismatch between users' queries and the way IR systems answer to such queries, and, depending on the situation, it can hinder the retrieval of relevant documents, affect the quality of the produced ranking list, or both. To investigate how the semantic gap affects retrieval models, we started with an in-depth evaluation of lexical and semantic signals. This allowed us to consider the problem from a comprehensive perspective and to understand the inherent complexity of addressing the semantic gap. The evaluation was performed through different analyses, which focused – from different angles – on semantic models and their relation with lexical models. The outcomes of these analyses highlighted the complementary nature of lexical and semantic signals, the need to combine them at the early stages of the IR pipeline to effectively address the semantic gap, and the semantic models that are best suited for the task. In particular, one of the analyses showed that semantic matching, when used to perform first-stage retrieval rather than re-ranking, allows models to retrieve relevant documents that are most affected by the semantic gap – thus motivating the development of semantic models effective at the early stages of the IR pipeline.

Based on the insights and lessons learned from the evaluation of lexical and semantic signals, we investigated the use of external knowledge resources to enhance lexical and semantic models and address the semantic gap. Specifically, we developed unsupervised knowledge-enhanced models, and we evaluated them in the medical domain. The medical domain presents characteristics that allowed us to explore the integration of external knowledge within unsupervised retrieval models. First, it is a domain where labeled data are scarce and expensive resources; secondly, it is a domain where the semantic gap is prominent; and thirdly, it is a domain rich of authoritative, manually curated by professionals, knowledge resources.

For lexical models, we conducted a series of studies and analyses exploring the use of external knowledge resources to enhance query representations in the context of precision medicine. In this regard, we developed several knowledge-based query reformulation techniques, and we tested them on the TREC PM Track.

At first, we proposed a procedure to expand queries iteratively and filter out trials for which a target patient is not eligible. This preliminary study served to understand whether retrieval performance can be correlated with the relational information used in the query expansion process. The experimental evaluation, conducted on the TREC PM 2018 Clinical Trials task, showed that without the use of an appropriate weighting scheme on query terms, as well as of knowledge resources tailored for the task, query expansion incurs in topic drift and leads to detrimental results.

Given the outcomes of this study, we deepened the investigation on effective ways of integrating external knowledge into query representations. To this end, we developed several knowledge-based query expansion and reduction techniques, and we explored their effectiveness in both TREC PM tasks: scientific literature and clinical trials retrieval. The objective of this in-depth analysis was twofold: evaluate the effectiveness of the proposed query reformulations, and investigate whether the two tasks share common characteristics that the developed query reformulations can grasp. We performed experiments on TREC PM 2017 and 2018 Tracks, and we found that the proposed query reformulations perform well in both tasks. However, the results highlighted different trends for the two tasks. In particular, knowledge-based query expansions proved effective for scientific literature retrieval, whereas knowledge-based query reductions for clinical trials retrieval.

Successively, we conducted a validation study on TREC PM 2019 to test the effectiveness of the developed query reformulation techniques. Although we performed experiments on both tasks, we focused on clinical trials retrieval. The experimental results highlighted the effectiveness of the tested query reformulations in retrieving relevant trials at top positions of the ranking list – thus proving the robustness of the developed techniques across the years. Moreover, the analysis also showed that different query reformulations provide top performances on different topics. This means that the developed query reformulations – by focusing on different (semantic) aspects of the queries – promote distinct information that can be used to improve retrieval performance. As a consequence, when appropriately combined, the proposed reformulations can be used to enhance lexical models from several angles.

Finally, given the effectiveness of the developed techniques for clinical trials retrieval, we performed an a posteriori analysis that helped to identify a subset of query reformulations robust to the different sets of topics provided by TREC PM across the years. The selected query reformulations can be used (and combined) by multi-stage IR systems to obtain a richer pool of relevant documents at the early stages of the IR pipeline – thus reducing the semantic gap between queries and documents.

Hence, throughout the studies and analyses conducted, we explored the use of external knowledge resources to enhance lexical models, and we developed effective, state-of-the-art knowledge-based query reformulations that help to reduce the semantic gap in precision medicine.

Regarding semantic models, we first performed an analysis of the knowledge-enhanced neural language models proposed in the literature. The analysis aimed to understand the critical aspects of such models and evaluate their effectiveness when used to perform retrieval. The experimental evaluation highlighted two main limitations of knowledge-enhanced neural language models: one is related to the way they are applied to IR, the other is related to their effectiveness in modeling IR tasks. In fact, knowledge-enhanced neural language models have been used in IR mostly to perform re-ranking – where the integration of external knowledge cannot express its full potential. On the other hand, (knowledge-enhanced) neural language models do not encode relevance signals or discriminative aspects between queries and documents, which are fundamental to effectively address IR tasks.

To overcome the above limitations, we developed SAFIR, which learns representations that are optimized for IR and encodes linguistic features relevant to address the semantic gap between queries and documents. The experimental evaluation we conducted to investigate SAFIR effectiveness at the early stages of the IR pipeline compared SAFIR with knowledge-enhanced neural language models on the TREC CDS Track considering two retrieval strategies: document retrieval and query expansion.

Document retrieval gave us the opportunity to investigate the effectiveness of integrating external knowledge into neural models for the typical retrieval scenario. In particular, the experimental evaluation showed that the integration of external knowledge into the learning process of neural IR models proves effective and helps to reduce the semantic gap between queries and documents. In this regard, the analysis of the results highlighted that SAFIR retrieves relevant documents that none of the knowledge-enhanced neural language models discover. Besides, the linguistic features encoded by SAFIR help to boost the results at the top positions of the ranking list.

On the other hand, query expansion allowed us to investigate the effectiveness of knowledge-enhanced neural models – which are specifically designed to address the semantic gap – when used to provide expansion terms that help lexical models to reduce the semantic gap. The outcomes of this evaluation showed that knowledge-enhanced neural models, and in particular SAFIR, grasp different signals than lexical models or corpus-driven neural models and retrieve documents in top positions from which better expansion terms can be extracted.

Thus, the in-depth analyses we performed to evaluate SAFIR highlighted its ability to address the semantic gap, as well as the effectiveness of combining lexical and semantic models at the early stages of the IR pipeline – where the complementary signals they provide can be used to obtain better answers to semantically hard queries.

A large portion of the research conducted in this thesis sets within a long-term research project [157], which is related to the ExaMode project.[1] ExaMode objective is to provide knowledge discovery tools for exascale multimodal medical data. In this context, we aim to continue working on the topics investigated in this thesis to develop cutting edge knowledge-enhanced retrieval systems for real-case CDS applications. For this reason, further investigation can be devoted to the following topics.

Given the complementary nature of lexical and semantic signals, as well as the effectiveness of integrating external knowledge into retrieval models, we plan to investigate the combination of the proposed knowledge-enhanced lexical and semantic models to develop multi-stage IR systems for CDS. Among the possible directions, it would be interesting to analyze the use of query performance predictors – based on query linguistic features – to decide, for instance, how to reformulate the given query or which SAFIR variant is more appropriate to perform retrieval. On the other hand, we could explore the use of early and late fusion techniques to combine multiple pre-retrieval knowledge-based query reformulations and build systems less sensitive to the problem of topic drift. Finally, the use of the different representations learned by SAFIR as input features to deep neural re-rankers would also be worth studying to understand the impact that knowledge-enhanced representations – optimized for IR – can have on the learning process of these models.

The medical domain presents a high heterogeneity, and both textual and visual data are often relevant for CDS applications. For this reason, the integration of visual features in the learning process of neural IR models can further enhance their understanding of the problem at hand – and also enable the retrieval of relevant information from one modality to the other. To this end, the recent advances in unsupervised visual representation learning [39], along with the successful applications of Transformer networks to the multimodal scenario [127, 150], opened up promising directions we could investigate to develop multimodal IR systems for CDS applications.

[1] https://www.examode.eu/

List of acronyms

ARC-II Convolutional Matching Model Architecture-II 38

BERT Bidirectional Encoder Representations for Transformers 30

BoWE Bag-of-Word-Embeddings 36

CBOW Continuous Bag-Of-Words 29

CDS Clinical Decision Support 5

cdoc2vec conceptual doc2vec 41

CLEF Conference and Labs of the Evaluation Forum 10

CLSM Convolutional Latent Semantic Model 38

CNN Convolutional Neural Network 38

Conv-KNRM Convolutional Kernel-based Neural Ranking Model 39

Co-PACRR Context-aware PACRR 39

CUI Concept Unique Identifier 17

DBOW Distributed Bag-Of-Words 30

DFR Divergence From Randomness 45

DM Distributed Memory 30

Doc2VecC Document Vector through Corruption 31

DRMM Deep Relevance Matching Model 38

DSSM Deep Structured Semantic Model 37

ECIR European Conference on Information Retrieval 44

EHR Electronic Health Records 11

EL Entity Linking 20

ELMo Embeddings from Language Models 30

FFNN Feed Forward Neural Network 38

FIRE Forum for Information Retrieval Evaluation 10

GloVe Global Vector 30

HiNT Hierarchical Neural maTching model 39

HPI	History of Present Illness	11
IDF	Inverse Document Frequency	36
IE	Information Extraction	19
IR	Information Retrieval	1
K-NRM	Kernel-based Neural Ranking Model	39
KDE	Kernel Density Estimation	50
KELSI	Knowledge-Enhanced LSI	40
KLD	Kullback-Leibler Divergence	50
LDA	Latent Dirichlet Allocation	34
LSE	Latent Semantic Entities	37
LSI	Latent Semantic Indexing	33
LSTM	Long Short-Term Memory	30
LSTM-DSSM	LSTM Deep Structured Semantic Model	38
MeSH	Medical Subject Headings	16
MLP	Multi Layer Perceptron	38
NCE	Noise Constrastive Estimation	131
NCI	National Cancer Institute	16
NER	Named Entity Recognition	20
NIST	National Institute of Standards and Technology	2
NLM	National Library of Medicine	16
NLP	Natural Language Processing	3
NTCIR	NII Testbeds and Community for Information access Research	10
NVSM	Neural Vector Space Model	37
PACRR	Position-Aware Convolutional-Recurrent Relevance Matching	39
PDF	Probability Density Function	50
PL2	Poisson estimation for randomness using Laplace succession for normalisation	46
pLSI	probabilistic LSI	34
PM	Precision Medicine	4
PMC	PubMed Central	11
PRF	Pseudo Relevance Feedback	22
QLM	Query Likelihood Model	38
RCM	Relation Constrained Model	128
rdoc2vec	retrofitted doc2vec	41
RF	Relevance Feedback	21

RNN Recurrent Neural Network 39

rword2vec retrofitted word2vec 31

SAFIR Semantic-Aware Neural Framework for IR 5

SIGIR ACM SIGIR Conference on Research & Development in Information Retrieval 43

SMART System for the Mechanical Analysis and Retrieval of Text 1

SNOMED CT Systematized Nomenclature of Medicine - Clinical Terms 15

SPECTER Scientific Paper Embeddings using Citation-informed TransformERs 31

SRNN Spatial Recurrent Neural Network 39

SVD Singular Value Decomposition 33

WSD Word Sense Disambiguation 20

S-WSD Shallow Word Sense Disambiguation 146

TREC Text REtrieval Conference 2

UMLS Unified Medical Language System 16

VF Vocabulary Feedback 21

WSD Word Sense Disambiguation 20

List of released resources

In this thesis, we have shown the importance of reproducibility as a means to understand and advance research in fields grounded into experimental evaluation, like Information Retrieval. For this reason, we release to the research community most of the resources developed for and presented in this thesis. In this way, we hope to ease future researchers further developing IR systems based on our findings – and also reproducing/exploiting our experiments/results.

Lexical and Semantic Signals

We release the data, source code, and plots for the in-depth evaluation of lexical and semantic models presented in Chapter 4 at https://github.com/giansilv/NeuralIR/.

Knowledge-Enhanced Lexical Models

We release the data used to perform the experiments for the different studies and analyses on knowledge-enhanced lexical models presented in Chapter 5 at https://github.com/stefano-marchesin/TREC_PM_qreforms/.

Knowledge-Enhanced Word Embeddings for IR

We release the source code used in the reproducibility study on knowledge-enhanced word embeddings presented in Section 6.2 at https://github.com/stefano-marchesin/learning_ke_wembs/.

Knowledge-Enhanced Document Embeddings for IR

We release the source code used in the reproducibility study on knowledge-enhanced document embeddings presented in Section 6.3 at https://github.com/stefano-marchesin/learning_ke_dembs/.

The Semantic-Aware Neural Framework for IR

We release the source code developed to perform the experiments with SAFIR in Sections 6.7 and 6.8 at https://github.com/stefano-marchesin/SAFIR/. Also, evaluation results and statistical analyses are publicly available at https://zenodo.org/record/3908196#.X2kbwWgzZPY.

References

[1] Agirre, E., Ansa, O., Arregi, X., de Lacalle, M. L., Otegi, A., and Saralegi, X. (2010a). Document Expansion for Cross-Lingual Passage Retrieval. In *CLEF 2010 LABs and Workshops, Notebook Papers, 22-23 September 2010, Padua, Italy*, volume 1176 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[2] Agirre, E., Arregi, X., and Otegi, A. (2010b). Document Expansion Based on WordNet for Robust IR. In *Proc. of the 23rd International Conference on Computational Linguistics, COLING 2010, 23-27 August 2010, Beijing, China*, pages 9–17. ACL.

[3] Agirre, E., Di Nunzio, G. M., Ferro, N., Mandl, T., and Peters, C. (2008). CLEF 2008: Ad hoc track overview. In *Evaluating Systems for Multilingual and Multimodal Information Access, 9th Workshop of the Cross-Language Evaluation Forum, CLEF 2008, Aarhus, Denmark, September 17-19, 2008, Revised Selected Papers*, volume 5706 of *Lecture Notes in Computer Science*, pages 15–37. Springer.

[4] Agosti, M., Di Nunzio, G. M., and Marchesin, S. (2018a). The University of Padua IMS Research Group at TREC 2018 Precision Medicine Track. In *Proc. of the Twenty-Seventh Text REtrieval Conference, TREC 2018, Gaithersburg, Maryland, USA, November 14-16, 2018*, volume 500-331 of *NIST Special Publication*. NIST.

[5] Agosti, M., Di Nunzio, G. M., and Marchesin, S. (2019a). An Analysis of Query Reformulation Techniques for Precision Medicine. In *Proc. of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 973–976. ACM.

[6] Agosti, M., Di Nunzio, G. M., and Marchesin, S. (2020a). A Post-Analysis of Query Reformulation Methods for Clinical Trials Retrieval. In *Proc. of the 28th Italian Symposium on Advanced Database Systems, Villasimius, Sud Sardegna, Italy (virtual due to Covid-19 pandemic), June 21-24, 2020*, volume 2646 of *CEUR Workshop Proceedings*, pages 152–159. CEUR-WS.org.

[7] Agosti, M., Di Nunzio, G. M., Marchesin, S., and Silvello, G. (2018b). A Relation Extraction Approach for Clinical Decision Support. In *Proc. of the CIKM 2018 Workshops co-located with 27th ACM International Conference on Information and Knowledge Management (CIKM 2018), Torino, Italy, October 22, 2018*, volume 2482 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[8] Agosti, M., Fabris, E., and Silvello, G. (2019b). On Synergies Between Information Retrieval and Digital Libraries. In *Proc. of the 15th Italian Research Conference on Digital Libraries, IRCDL 2019, Pisa, Italy, January 31 - February 1, 2019*, volume 988 of *Communications in Computer and Information Science*, pages 3–17. Springer.

[9] Agosti, M., Marchesin, S., and Silvello, G. (2020b). Learning Unsupervised Knowledge-Enhanced Representations to Reduce the Semantic Gap in Information Retrieval. *ACM Trans. Inf. Syst.*, 38(4).

[10] Ai, Q., Yang, L., Guo, J., and Croft, W. B. (2016a). Analysis of the Paragraph Vector Model for Information Retrieval. In *Proc. of the 2016 ACM International Conference on the Theory of Information Retrieval, ICTIR 2016, Newark, DE, USA, September 12-6, 2016*, pages 133–142. ACM.

[11] Ai, Q., Yang, L., Guo, J., and Croft, W. B. (2016b). Improving Language Estimation with the Paragraph Vector Model for Ad-hoc Retrieval. In *Proc. of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, pages 869–872. ACM.

[12] Allan, J., Harman, D., Kanoulas, E., Li, D., Van Gysel, C., and Voorhees, E. M. (2017). TREC 2017 common core track overview. In *Proc. of The Twenty-Sixth Text REtrieval Conference, TREC 2017, Gaithersburg, Maryland, USA, November 15-17, 2017*, volume 500-324 of *NIST Special Publication*. NIST.

[13] Amati, G. and van Rijsbergen, C. J. (2002). Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389.

[14] Arguello, J., Crane, M., Diaz, F., Lin, J., and Trotman, A. (2015). Report on the SIGIR 2015 Workshop on Reproducibility, Inexplicability, and Generalizability of Results (RIGOR). *SIGIR Forum*, 49(2):107–116.

[15] Aronson, A. R. (2001). Effective mapping of biomedical text to the umls metathesaurus: the metamap program. In *Proceedings of the AMIA Symposium*, page 17. American Medical Informatics Association.

[16] Aronson, A. R. and Lang, F. M. (2010). An Overview of MetaMap: Historical Perspective and Recent Advances. *J. Am. Medical Informatics Assoc.*, 17(3):229–236.

[17] Aronson, A. R. and Rindflesch, T. C. (1997). Query expansion using the UMLS Metathesaurus. In *Proc. of the American Medical Informatics Association Annual Symposium, AMIA 1997, Nashville, TN, USA, October 25-29, 1997*. AMIA.

[18] Attar, R. and Fraenkel, A. S. (1977). Local Feedback in Full-Text Retrieval Systems. *J. ACM*, 24(3):397–417.

[19] Baeza-Yates, R. A. and Ribeiro-Neto, B. A. (1999). *Modern Information Retrieval*. ACM Press / Addison-Wesley.

[20] Balog, K., Azzopardi, L., and de Rijke, M. (2006). Formal Models for Expert Finding in Enterprise Corpora. In *Proc. of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006*, pages 43–50. ACM.

[21] Balog, K., Azzopardi, L., and de Rijke, M. (2009). A Language Modeling Framework for Expert Finding. *Inf. Process. Manag.*, 45(1):1–19.

[22] Bean, C. A. and Green, R. (2001). *Relationships in the Organization of Knowledge*. Information Science and Knowledge Management 2. Springer, 1 edition.

[23] Belkin, N. J., Oddy, R. N., and Brooks, H. M. (1982). ASK for Information Retrieval: Part I. Background and Theory. *Journal of Documentation*, 38(2):61–71.

[24] Bengio, Y., Courville, A. C., and Vincent, P. (2013). Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828.

[25] Bengio, Y., Delalleau, O., and Roux, N. L. (2006). Label Propagation and Quadratic Criterion. In *Semi-Supervised Learning*, pages 192–216. The MIT Press.

[26] Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A Neural Probabilistic Language Model. *J. of Mach. Learn. Res.*, 3:1137–1155.

[27] Berger, A. L. and Lafferty, J. D. (1999). Information Retrieval as Statistical Translation. In *Proc. of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 15-19, 1999, Berkeley, CA, USA*, pages 222–229. ACM.

[28] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3:993–1022.

[29] Bodenreider, O. (2004). The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270.

[30] Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguistics*, 5:135–146.

[31] Booth, A. and O'Rourke, A. (1997). The value of structured abstracts in information retrieval from MEDLINE. *Health Libraries Review*, 14(3):157–166.

[32] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language Models are Few-Shot Learners. *CoRR*, abs/2005.14165.

[33] Buckley, C., Singhal, A., and Mitra, M. (1995). New Retrieval Approaches Using SMART: TREC 4. In *Proc. of The Fourth Text REtrieval Conference, TREC 1995, Gaithersburg, Maryland, USA, November 1-3, 1995*, volume 500-236 of *NIST Special Publication*. NIST.

[34] Burnham, K. P. and Anderson, D. R. (1998). Practical Use of the Information-Theoretic Approach. In *Model Selection and Inference*, pages 75–117. Springer.

[35] Bush, V. (1945). As We May Think. *The Atlantic Monthly*, 176(1):101–108.

[36] Callan, J. P., Croft, W. B., and Broglio, J. (1995). TREC and Tipster Experiments with Inquery. *Inf. Process. Manag.*, 31(3):327–343.

[37] Carterette, B. A. (2012). Multiple Testing in Statistical Analysis of Systems-Based Information Retrieval Experiments. *ACM Trans. Inf. Syst.*, 30(1):4:1–4:34.

[38] Chen, M. (2017). Efficient Vector Representation for Documents through Corruption. In *Proc. of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017*. OpenReview.net.

[39] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. E. (2020). A Simple Framework for Contrastive Learning of Visual Representations. *CoRR*, abs/2002.05709.

[40] Cheng, J., Wang, Z., Wen, J. R., Yan, J., and Chen, Z. (2015). Contextual Text Understanding in Distributional Semantic Space. In *Proc. of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, pages 133–142. ACM.

[41] Choi, E., Bahadori, M. T., Searles, E., Coffey, C., Thompson, M., Bost, J., Tejedor-Sojo, J., and Sun, J. (2016). Multi-layer Representation Learning for Medical Concepts. In *Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1495–1504. ACM.

[42] Clancy, R., Ferro, N., Hauff, C., Lin, J., Sakai, T., and Wu, Z. Z. (2019). The SIGIR 2019 Open-Source IR Replicability Challenge (OSIRRC 2019). In *Proc. of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 1432–1434. ACM.

[43] Cleverdon, C. (1967). The Cranfield Tests on Index Language Devices. In *Proc. Aslib*, volume 19, pages 173–194. MCB UP Ltd.

[44] Cleverdon, C. W. (1991). The Significance of the Cranfield Tests on Index Languages. In *Proc. of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Chicago, Illinois, USA, October 13-16, 1991 (Special Issue of the SIGIR Forum)*, pages 3–12. ACM.

[45] Cohan, A., Feldman, S., Beltagy, I., Downey, D., and Weld, D. S. (2020). SPECTER: Document-level Representation Learning using Citation-informed Transformers. In *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2270–2282. ACL.

[46] Cover, T. M. and Thomas, J. A. (2012). *Elements of Information Theory*. John Wiley & Sons.

[47] Craswell, N., Croft, W. B., de Rijke, M., Guo, J., and Mitra, B. (2017). SIGIR 2017 workshop on neural information retrieval (neu-ir'17). In *Proc. of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 1431–1432. ACM.

[48] Craswell, N., Croft, W. B., de Rijke, M., Guo, J., and Mitra, B. (2018). Neural information retrieval: introduction to the special issue. *Inf. Retr. J.*, 21(2-3):107–110.

[49] Craswell, N., Croft, W. B., Guo, J., Mitra, B., and de Rijke, M. (2016). Neu-ir: The SIGIR 2016 workshop on neural information retrieval. In *Proc. of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, pages 1245–1246. ACM.

[50] Craswell, N., Mitra, B., Yilmaz, E., Campos, D., and Voorhees, E. M. (2020). Overview of the TREC 2019 deep learning track. *CoRR*, abs/2003.07820.

[51] Crestani, F. (2000). Exploiting the Similarity of Non-Matching Terms at Retrieval Time. *Inf. Retr.*, 2(1):23–43.

[52] Croft, W. B. and Harper, D. J. (1979). Using Probabilistic Models of Document Retrieval without Relevance Information. *Journal of Documentation*, 35(4):285–295.

[53] Dai, Z., Xiong, C., Callan, J., and Liu, Z. (2018). Convolutional Neural Networks for Soft-Matching N-Grams in Ad-hoc Search. In *Proc. of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, pages 126–134. ACM.

[54] Das, D. and Petrov, S. (2011). Unsupervised Part-of-Speech Tagging with Bilingual Graph-Based Projections. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 19-24 June, 2011, Portland, Oregon, USA*, pages 600–609. ACL.

[55] Das, D. and Smith, N. A. (2011). Semi-Supervised Frame-Semantic Parsing for Unknown Predicates. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 19-24 June, 2011, Portland, Oregon, USA*, pages 1435–1444. ACL.

[56] De Vine, L., Zuccon, G., Koopman, B., Sitbon, L., and Bruza, P. (2014). Medical Semantic Similarity with a Neural Language Model. In *Proc. of the 23rd ACM International Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, pages 1819–1822. ACM.

[57] Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A. (1990). Indexing by Latent Semantic Analysis. *J. Am. Soc. Inf. Sci.*, 41(6):391–407.

[58] Demartini, G., Gaugaz, J., and Nejdl, W. (2009). A Vector Space Model for Ranking Entities and Its Application to Expert Search. In *Proc. of the 31st European Conference on IR Research, ECIR 2009, Toulouse, France, April 6-9, 2009*, volume 5478 of *Lecture Notes in Computer Science*, pages 189–201. Springer.

[59] Devlin, J., Chang, M. W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019*, pages 4171–4186. ACL.

[60] Di Nunzio, G. M., Ferro, N., Mandl, T., and Peters, C. (2007). CLEF 2007: Ad hoc track overview. In *Proc. of the 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Budapest, Hungary, September 19-21, 2007*, volume 5152 of *Lecture Notes in Computer Science*, pages 13–32. Springer.

[61] Di Nunzio, G. M., Marchesin, S., and Agosti, M. (2019). Exploring how to Combine Query Reformulations for Precision Medicine. In *Proc. of the Twenty-Eighth Text REtrieval Conference, TREC 2019, Gaithersburg, Maryland, USA, November 13-15, 2019*, volume 1250 of *NIST Special Publication*. NIST.

[62] Diao, L., Yan, H., Li, F., Song, S., Lei, G., and Wang, F. (2018). The Research of Query Expansion Based on Medical Terms Reweighting in Medical Information Retrieval. *EURASIP J. Wireless Comm. and Networking*, 2018:105.

[63] Diaz, F., Mitra, B., and Craswell, N. (2016). Query Expansion with Locally-Trained Word Embeddings. In *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*. ACL.

[64] Dienstmann, R., Jang, I. S., Bot, B., Friend, S., and Guinney, J. (2015). Database of Genomic Biomarkers for Cancer Drugs and Clinical Targetability in Solid Tumors. *Cancer Discov.*, 5(2):118–123.

[65] Dinh, D. and Tamine, L. (2012). Towards a Context Sensitive Approach to Searching Information Based on Domain Specific Knowledge Sources. *J. Web Semant.*, 12:41–52.

[66] Donnelly, K. (2006). SNOMED-CT: The advanced terminology and coding system for eHealth. *Studies in health technology and informatics*, 121:279.

[67] Duchi, J. C., Hazan, E., and Singer, Y. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *J. Mach. Learn. Res.*, 12:2121–2159.

[68] Dumais, S. T. (1994). Latent Semantic Indexing (LSI): TREC-3 Report. In *Proc. of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*, volume 500-225 of *NIST Special Publication*, pages 219–230. NIST.

[69] Dür, A., Rauber, A., and Filzmoser, P. (2018). Reproducing a Neural Question Answering Architecture Applied to the SQuAD Benchmark Dataset: Challenges and Lessons Learned. In *Proc. of the 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018*, volume 10772 of *Lecture Notes in Computer Science*, pages 102–113. Springer.

[70] Edinger, T., Cohen, A. M., Bedrick, S., Ambert, K. H., and Hersh, W. R. (2012). Barriers to Retrieving Patient Information from Electronic Health Record Data: Failure Analysis from the TREC Medical Records Track. In *AMIA 2012, American Medical Informatics Association Annual Symposium, Chicago, Illinois, USA, November 3-7, 2012*. AMIA.

[71] Faessler, E., Hahn, U., and Oleynik, M. (2019). JULIE Lab & Med Uni Graz @ TREC 2019 Precision Medicine Track. In *Proc. of the Twenty-Eighth Text REtrieval Conference, TREC 2019, Gaithersburg, Maryland, USA, November 13-15, 2019*, volume 1250 of *NIST Special Publication*. National Institute of Standards and Technology (NIST).

[72] Fan, Y., Guo, J., Lan, Y., Xu, J., Zhai, C., and Cheng, X. (2018). Modeling Diverse Relevance Patterns in Ad-hoc Retrieval. In *Proc. of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 375–384. ACM.

[73] Fang, H. (2008). A Re-examination of Query Expansion Using Lexical Resources. In *Proc. of the 46th Annual Meeting of the Association for Computational Linguistics, ACL 2008, June 15-20, 2008, Columbus, Ohio, USA*, pages 139–147. ACL.

[74] Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., and Smith, N. A. (2015). Retrofitting Word Vectors to Semantic Lexicons. In *Proc. of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1606–1615. ACL.

[75] Ferragina, P. and Scaiella, U. (2010). TAGME: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proc. of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, pages 1625–1628. ACM.

[76] Ferro, N., Fuhr, N., Järvelin, K., Kando, N., Lippold, M., and Zobel, J. (2016). Increasing Reproducibility in IR: Findings from the Dagstuhl Seminar on "Reproducibility of Data-Oriented Experiments in e-Science". *SIGIR Forum*, 50(1):68–82.

[77] Ferro, N., Fuhr, N., and Rauber, A. (2018a). Introduction to the Special Issue on Reproducibility in Information Retrieval: Evaluation Campaigns, Collections, and Analyses. *J. Data and Information Quality*, 10(3):9:1–9:4.

[78] Ferro, N., Fuhr, N., and Rauber, A. (2018b). Introduction to the Special Issue on Reproducibility in Information Retrieval: Tools and Infrastructures. *J. Data and Information Quality*, 10(4):14:1–14:4.

[79] Ferro, N., Marchesin, S., Purpura, A., and Silvello, G. (2019). A Docker-Based Replicability Study of a Neural Information Retrieval Model. In *Proc. of the Open-Source IR Replicability Challenge co-located with 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, OSIRRC@SIGIR 2019, Paris, France, July 25, 2019*, volume 2409 of *CEUR Workshop Proceedings*, pages 37–43. CEUR-WS.org.

[80] Ferro, N., Marchesin, S., Purpura, A., and Silvello, G. (2020). Reproducibility of the Neural Vector Space Model via Docker. In *Proc. of the 16th Italian Research Conference on Digital Libraries, IRCDL 2020, Bari, Italy, January 30-31, 2020*, volume 1177 of *Communications in Computer and Information Science*, pages 3–8. Springer.

[81] Ferro, N. and Peters, C. (2009). CLEF 2009 Ad Hoc Track Overview: TEL and Persian Tasks. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments, 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Corfu, Greece, September 30 - October 2, 2009, Revised Selected Papers*, volume 6241 of *Lecture Notes in Computer Science*, pages 13–35. Springer.

[82] Ferro, N. and Silvello, G. (2015). Rank-Biased Precision Reloaded: Reproducibility and Generalization. In *Proc. of the 37th European Conference on IR Research, ECIR 2015, Vienna, Austria, March 29 - April 2, 2015*, volume 9022 of *Lecture Notes in Computer Science*, pages 768–780.

[83] Fu, G., Jones, C. B., and Abdelmoty, A. I. (2005). Ontology-Based Spatial Query Expansion in Information Retrieval. In *Proc. of On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE, OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2005, Agia Napa, Cyprus, October 31 - November 4, 2005, Part II*, volume 3761 of *Lecture Notes in Computer Science*, pages 1466–1482. Springer.

[84] Fuhr, N. (2017). Some Common Mistakes In IR Evaluation, And How They Can Be Avoided. *SIGIR Forum*, 51(3):32–41.

[85] Furnas, G. W., Landauer, T. K., Gomez, L. M., and Dumais, S. T. (1987). The Vocabulary Problem in Human-System Communication. *Commun. ACM*, 30(11):964–971.

[86] Ganguly, D., Roy, D., Mitra, M., and Jones, G. J. F. (2015). Word Embedding based Generalized Language Model for Information Retrieval. In *Proc. of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, pages 795–798. ACM.

[87] Glavaš, G. and Vulić, I. (2018). Explicit Retrofitting of Distributional Word Vectors. In *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018*, pages 34–45. ACL.

[88] Glorot, X. and Bengio, Y. (2010). Understanding the Difficulty of Training Deep Feedforward Neural Networks. In *Proc. of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, volume 9 of *JMLR Proceedings*, pages 249–256. JMLR.org.

[89] Goeuriot, L., Jones, G. J. F., Kelly, L., Müller, H., and Zobel, J. (2016). Medical Information Retrieval: Introduction to the Special Issue. *Inf. Retr. J.*, 19(1-2):1–5.

[90] Goodwin, T. R., Skinner, M. A., and Harabagiu, S. M. (2017). UTD HLTRI at TREC 2017: Precision Medicine Track. In *Proc. of The Twenty-Sixth Text REtrieval Conference, TREC 2017, Gaithersburg, Maryland, USA, November 15-17, 2017*, volume 500-324 of *NIST Special Publication*. NIST.

[91] Gruber, T. R. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2):199–221.

[92] Guarino, N., Oberle, D., and Staab, S. (2009). What is an ontology? In *Handbook on Ontologies*, International Handbooks on Information Systems, pages 1–17. Springer.

[93] Gülçehre, Ç., Moczulski, M., Denil, M., and Bengio, Y. (2016). Noisy Activation Functions. In *Proc. of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 3059–3068. JMLR.org.

[94] Guo, D., Berry, M. W., Thompson, B. B., and Bailin, S. C. (2003). Knowledge-Enhanced Latent Semantic Indexing. *Inf. Retr.*, 6(2):225–250.

[95] Guo, J., Fan, Y., Ai, Q., and Croft, W. B. (2016a). A Deep Relevance Matching Model for Ad-hoc Retrieval. In *Proc. of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, pages 55–64. ACM.

[96] Guo, J., Fan, Y., Ai, Q., and Croft, W. B. (2016b). Semantic Matching by Non-Linear Word Transportation for Information Retrieval. In *Proc. of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, pages 701–710. ACM.

[97] Gurulingappa, H., Toldo, L., Schepers, C., Bauer, A., and Megaro, G. (2016). Semi-Supervised Information Retrieval System for Clinical Decision Support. In *Proc. of The Twenty-Fifth Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*, volume 500-321 of *NIST Special Publication*. NIST.

[98] Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proc. of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, pages 297–304. JMLR.org.

[99] Harman, D. (1992a). The DARPA TIPSTER project. *SIGIR Forum*, 26(2):26–28.

[100] Harman, D. (1992b). Overview of the First Text REtrieval Conference (TREC-1). In *Proc. of The First Text REtrieval Conference, TREC 1992, Gaithersburg, Maryland, USA, November 4-6, 1992*, volume 500-207 of *NIST Special Publication*, pages 1–20. NIST.

[101] Harman, D. (1993). Document detection data preparation. In *TIPSTER TEXT PROGRAM: PHASE I: Proc. of a Workshop held at Fredricksburg, VA, USA, September 19-23, 1993*, pages 17–31. Morgan Kaufmann.

[102] Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

[103] Hasibi, F., Balog, K., and Bratsberg, S. E. (2016). Exploiting Entity Linking in Queries for Entity Retrieval. In *Proc. of the 2016 ACM International Conference on the Theory of Information Retrieval, ICTIR 2016, Newark, DE, USA, September 12- 6, 2016*, pages 209–218. ACM.

[104] Hawking, D., Voorhees, E. M., Craswell, N., and Bailey, P. (1999). Overview of the TREC-8 web track. In *Proc. of The Eighth Text REtrieval Conference, TREC 1999, Gaithersburg, Maryland, USA, November 17-19, 1999*, volume 500-246 of *NIST Special Publication*. NIST.

[105] Hersh, W. R. (2009). *Information Retrieval: A Health and Biomedical Perspective*. Health and Informatics Series. Springer.

[106] Hersh, W. R., Bhupatiraju, R. T., Ross, L., Cohen, A. M., Kraemer, D., and Johnson, P. (2004). TREC 2004 Genomics Track Overview. In *Proc. of the Thirteenth Text REtrieval Conference, TREC 2004, Gaithersburg, Maryland, USA, November 16-19, 2004*, volume 500-261 of *NIST Special Publication*. NIST.

[107] Hersh, W. R., Buckley, C., Leone, T. J., and Hickam, D. (1994). OHSUMED: An Interactive Retrieval Evaluation and New Large Test Collection for Research. In *Proc. of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland, 3-6 July 1994 (Special Issue of the SIGIR Forum)*, pages 192–201. ACM/Springer.

[108] Hersh, W. R., Cohen, A. M., Roberts, P. M., and Rekapalli, H. K. (2006). TREC 2006 Genomics Track Overview. In *Proc. of the Fifteenth Text REtrieval Conference, TREC 2006, Gaithersburg, Maryland, USA, November 14-17, 2006*, volume 500-272 of *NIST Special Publication*. NIST.

[109] Hersh, W. R., Cohen, A. M., Yang, J., Bhupatiraju, R. T., Roberts, P. M., and Hearst, M. A. (2005). TREC 2005 Genomics Track Overview. In *Proc. of the Fourteenth Text REtrieval Conference, TREC 2005, Gaithersburg, Maryland, USA, November 15-18, 2005*, volume 500-266 of *NIST Special Publication*. NIST.

[110] Hersh, W. R., Price, S., and Donohoe, L. (2000). Assessing Thesaurus-based Query Expansion Using the UMLS Metathesaurus. In *Proc. of the American Medical Informatics Association Annual Symposium, AMIA 2000, Los Angeles, CA, USA, November 4-8, 2000*. AMIA.

[111] Hinton, G. E. (1986). Learning Distributed Representations of Concepts. In *Proc. of the 8th Annual Conference of the Cognitive Science Society, Amherst, Massachusetts, USA, 1986*, volume 1, pages 1–12. Erlbaum.

[112] Hinton, G. E., Osindero, S., and Teh, Y. W. (2006). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7):1527–1554.

[113] Hofmann, T. (1999). Probabilistic Latent Semantic Indexing. In *Proc. of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 15-19, 1999, Berkeley, CA, USA*, pages 50–57. ACM.

[114] Hopfgartner, F., Hanbury, A., Müller, H., Eggel, I., Balog, K., Brodt, T., Cormack, G. V., Lin, J., Kalpathy-Cramer, J., Kando, N., Kato, M. P., Krithara, A., Gollub, T., Potthast, M., Viegas, E., and Mercer, S. (2018). Evaluation-as-a-Service for the Computational Sciences: Overview and Outlook. *J. Data and Information Quality*, 10(4):15:1–15:32.

[115] Hu, B., Lu, Z., Li, H., and Chen, Q. (2014). Convolutional Neural Network Architectures for Matching Natural Language Sentences. In *Proc. of the 27th Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2042–2050.

[116] Huang, P., He, X., Gao, J., Deng, L., Acero, A., and Heck, L. P. (2013). Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data. In *Proc. of the 22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, pages 2333–2338. ACM.

[117] Hui, K., Yates, A., Berberich, K., and de Melo, G. (2017). PACRR: A Position-Aware Neural IR Model for Relevance Matching. In *Proc. of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1049–1058. ACL.

[118] Hui, K., Yates, A., Berberich, K., and de Melo, G. (2018). Co-PACRR: A Context-Aware Neural IR Model for Ad-hoc Retrieval. In *Proc. of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, pages 279–287. ACM.

[119] Iacobacci, I., Pilehvar, M. T., and Navigli, R. (2015). SensEmbed: Learning Sense Embeddings for Word and Relational Similarity. In *Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China*, pages 95–105. ACL.

[120] Ioffe, S. and Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proc. of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 448–456. JMLR.org.

[121] Jaleel, N. A., Allan, J., Croft, W. B., Diaz, F., Larkey, L. S., Li, X., Smucker, M. D., and Wade, C. (2004). UMass at TREC 2004: Novelty and HARD. In *Proc. of the Thirteenth Text REtrieval Conference, TREC 2004, Gaithersburg, Maryland, USA, November 16-19, 2004*. NIST.

[122] Jimmy, Zuccon, G., and Koopman, B. (2018). QUT IElab at CLEF 2018 Consumer Health Search Task: Knowledge Base Retrieval for Consumer Health Search. In *Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018*, volume 2125 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[123] Johansson, R. and Piña, L. N. (2015). Embedding a Semantic Network in a Word Space. In *Proc. of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1428–1433. ACL.

[124] Jones, K. S. and Rijsbergen, C. V. (1975). *Report on the Need for and Provision of an 'ideal' Information Retrieval Test Collection*, volume 5266. University Computer Laboratory.

[125] Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling Laws for Neural Language Models. *CoRR*, abs/2001.08361.

[126] Kenter, T., Borisov, A., and de Rijke, M. (2016). Siamese CBOW: Optimizing Word Embeddings for Sentence Representations. In *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany*, pages 941–951. ACL.

[127] Kiela, D., Bhooshan, S., Firooz, H., and Testuggine, D. (2019). Supervised Multimodal Bitransformers for Classifying Images and Text. In *Proc. of the Visually Grounded Interaction and Language (ViGIL), NeurIPS 2019 Workshop, Vancouver, Canada, December 13, 2019*.

[128] Kingma, D. P. and Ba, J. (2015). Adam: A Method for Stochastic Optimization. In *Proc. of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015*.

[129] Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Skip-Thought Vectors. In *Proc. of the 29th Annual Conference on Neural Information Processing Systems 2015, 7-12 December, 2015, Montreal, Quebec, Canada*, pages 3294–3302.

[130] Koopman, B. and Zuccon, G. (2014). Why Assessing Relevance in Medical IR is Demanding. In *Proc. of the Medical Information Retrieval Workshop at SIGIR co-located with the 37th annual international ACM SIGIR conference (ACM SIGIR 2014), Gold Coast, Australia, July 11, 2014*, volume 1276 of *CEUR Workshop Proceedings*, pages 16–19. CEUR-WS.org.

[131] Koopman, B., Zuccon, G., Bruza, P., Sitbon, L., and Lawley, M. (2016). Information retrieval as semantic inference: a Graph Inference model applied to medical search. *Inf. Retr. Journal*, 19(1-2):6–37.

[132] Koopman, B., Zuccon, G., Nguyen, A. N., Vickers, D., Butt, L., and Bruza, P. (2012). Exploiting SNOMED CT Concepts & Relationships for Clinical Information Retrieval: Australian e-Health Research Centre and Queensland University of Technology at the TREC 2012 Medical Track. In *Proc. of The Twenty-First Text REtrieval Conference, TREC 2012, Gaithersburg, Maryland, USA, November 6-9, 2012*, volume 500-298 of *NIST Special Publication*. NIST.

[133] Krovetz, R. (1993). Viewing Morphology as an Inference Process. In *Proc. of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Pittsburgh, PA, USA, June 27 - July 1, 1993*, pages 191–202. ACM.

[134] Kullback, S. and Leibler, R. A. (1951). On Information and Sufficiency. *Ann. Math. Statist.*, 22(1):79–86.

[135] Kumaran, G. and Carvalho, V. R. (2009). Reducing Long Queries Using Query Quality Predictors. In *Proc. of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*, pages 564–571. ACM.

[136] Kuzi, S., Shtok, A., and Kurland, O. (2016). Query Expansion Using Word Embeddings. In *Proc. of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016*, pages 1929–1932. ACM.

[137] Lavrenko, V. and Croft, W. B. (2001). Relevance-Based Language Models. In *Proc. of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2001, New Orleans, Louisiana, USA, September 9-13, 2001*, pages 120–127. ACM.

[138] Le, Q. and Mikolov, T. (2014). Distributed Representations of Sentences and Documents. In *Proc. of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1188–1196. JMLR.org.

[139] Levy, O. and Goldberg, Y. (2014). Neural Word Embedding as Implicit Matrix Factorization. In *Proc. of the 28th Annual Conference on Neural Information Processing Systems 2014, 8-13 December, 2014, Montreal, Quebec, Canada*, pages 2177–2185.

[140] Li, H. and Xu, J. (2014). Semantic Matching in Search. *Found. Trends Inf. Retr.*, 7(5):343–469.

[141] Li, X., Guo, C., Chu, W., Wang, Y., and Shavlik, J. (2014). Deep Learning Powered in-Session Contextual Ranking Using Clickthrough Data. In *Proc. of the Workshop on personalization: Methods and applications co-located with the 28th Annual Conference on Neural Information Processing Systems 2014, 8-13 December, 2014, Montreal, Quebec, Canada*.

[142] Limsopatham, N., Macdonald, C., and Ounis, I. (2013a). A Task-Specific Query and Document Representation for Medical Records Search. In *Proc. of the 35th European Conference on IR Research, ECIR 2013, Moscow, Russia, March 24-27, 2013*, volume 7814 of *Lecture Notes in Computer Science*, pages 747–751. Springer.

[143] Limsopatham, N., Macdonald, C., and Ounis, I. (2013b). Inferring Conceptual Relationships to Improve Medical Records Search. In *Open research Areas in Information Retrieval, OAIR '13, Lisbon, Portugal, May 15-17, 2013*, pages 1–8. ACM.

[144] Lin, J. (2018). The Neural Hype and Comparisons Against Weak Baselines. *SIGIR Forum*, 52(2):40–51.

[145] Lin, J., Crane, M., Trotman, A., Callan, J., Chattopadhyaya, I., Foley, J., Ingersoll, G., MacDonald, C., and Vigna, S. (2016). Toward Reproducible Baselines: The Open-Source IR Reproducibility Challenge. In *Proc. of the 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016.*, volume 9626 of *Lecture Notes in Computer Science*, pages 408–420. Springer.

[146] Lipscomb, C. E. (2000). Medical Subject Headings (MeSH). *Bulletin of the Medical Library Association*, 88(3):265.

[147] Liu, X., Nie, J. Y., and Sordoni, A. (2016). Constraining Word Embeddings by Prior Knowledge - Application to Medical Information Retrieval. In *Proc. of the 12th Asia Information Retrieval Societies Conference, AIRS 2016, Beijing, China, November 30 - December 2, 2016*, pages 155–167. Springer.

[148] López-García, P., Oleynik, M., Kasác, Z., and Schulz, S. (2017). TREC 2017 Precision Medicine - Medical University of Graz. In *Proc. of The Twenty-Sixth Text REtrieval Conference, TREC 2017, Gaithersburg, Maryland, USA, November 15-17, 2017*, volume 500-324 of *NIST Special Publication*. NIST.

[149] Lovins, J. B. (1968). Development of a Stemming Algorithm. *Mech. Transl. Comput. Linguistics*, 11(1-2):22–31.

[150] Lu, J., Batra, D., Parikh, D., and Lee, S. (2019). ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *Proc. of the 33rd Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 13–23.

[151] Luhn, H. P. (1957). A Statistical Approach to Mechanized Encoding and Searching of Literary Information. *IBM J. Res. Dev.*, 1(4):309–317.

[152] Lv, Y. and Zhai, C. (2011). When Documents are Very Long, BM25 Fails! In *Proc. of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011*, pages 1103–1104. ACM.

[153] MacAvaney, S., Yates, A., Cohan, A., and Goharian, N. (2019). CEDR: Contextualized Embeddings for Document Ranking. In *Proc. of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 1101–1104. ACM.

[154] Macdonald, C., McCreadie, R., Santos, R. L. T., and Ounis, I. (2012). From Puppy to Maturity: Experiences in Developing Terrier. In *Proc. of the SIGIR 2012 Workshop on Open Source Information Retrieval, OSIR@SIGIR 2012, Portland, Oregon, USA, 16th August 2012*, pages 60–63. University of Otago, Dunedin, New Zealand.

[155] Mahmood, A. S. M. A., Li, G., Rao, S., McGarvey, P. B., Wu, C. H., Madhavan, S., and Vijay-Shanker, K. (2017). UD_GU_BioTM at TREC 2017: Precision Medicine Track. In *Proc. of The Twenty-Sixth Text REtrieval Conference, TREC 2017, Gaithersburg, Maryland, USA, November 15-17, 2017*, volume 500-324 of *NIST Special Publication*. NIST.

[156] Mancini, M., Camacho-Collados, J., Iacobacci, I., and Navigli, R. (2017). Embedding Words and Senses Together via Joint Knowledge-Enhanced Training. In *Proc. of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017*, pages 100–111. ACL.

[157] Marchesin, S. (2018). Case-Based Retrieval Using Document-Level Semantic Networks. In *Proc. of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, page 1451. ACM.

[158] Marchesin, S., Purpura, A., and Silvello, G. (2019). Focal Elements of Neural Information Retrieval Models. An Outlook through a Reproducibility Study. *Inf. Process. Manag.*, page 102109.

[159] Melamud, O., Goldberger, J., and Dagan, I. (2016). context2vec: Learning Generic Context Embedding with Bidirectional LSTM. In *Proc. of the 20th Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 51–61. ACL.

[160] Mendes, P. N., Jakob, M., García-Silva, A., and Bizer, C. (2011). DBpedia Spotlight: Shedding Light on the Web of Documents. In *Proc. of the 7th International Conference on Semantic Systems, I-SEMANTICS 2011, Graz, Austria, September 7-9, 2011*, ACM International Conference Proceeding Series, pages 1–8. ACM.

[161] Mihalcea, R. and Moldovan, D. (2000). Semantic Indexing using WordNet Senses. In *Proc. of the ACL-2000 Workshop on Recent Advances in Natural Language Processing and Information Retrieval. Hong Kong, China, October 2000*, pages 35–45. ACL.

[162] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

[163] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Proc. of the 27th Annual Conference on Neural Information Processing Systems 2013, 5-8 December, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.

[164] Miller, G. A. (1995). WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41.

[165] Mitra, B. and Craswell, N. (2018). An Introduction to Neural Information Retrieval. *Found. Trends Inf. Retr.*, 13(1):1–126.

[166] Mrkšic, N., OSéaghdha, D., Thomson, B., Gašic, M., Rojas-Barahona, L., Su, P. H., Vandyke, D., Wen, T. H., and Young, S. (2016). Counter-fitting Word Vectors to Linguistic Constraints. In *Proc. of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 142–148. ACL.

[167] Navigli, R. and Velardi, P. (2003). An Analysis of Ontology-based Query Expansion Strategies. In *Proc. of the International Workshop & Tutorial on Adaptive Text Extraction and Mining, held in conjunction with the 14th European Conference on Machine Learning and the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 42–49.

[168] Nguyen, G., Soulier, L., Tamine, L., and Bricon-Souf, N. (2017a). DSRIM: A Deep Neural Information Retrieval Model Enhanced by a Knowledge Resource Driven Representation of Documents. In *Proc. of the ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR 2017, Amsterdam, The Netherlands, October 1-4, 2017*, pages 19–26. ACM.

[169] Nguyen, G. H., Tamine, L., Soulier, L., and Souf, N. (2017b). Learning Concept-Driven Document Embeddings for Medical Information Search. In *Proc. of the 16th Conference on Artificial Intelligence in Medicine, AIME 2017, Vienna, Austria, June 21-24, 2017*, pages 160–170. Springer.

[170] Nguyen, G. H., Tamine, L., Soulier, L., and Souf, N. (2018). A Tri-Partite Neural Document Language Model for Semantic Information Retrieval. In *Proc. of the 15th European Semantic Web Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018*, pages 445–461. Springer.

[171] Nogueira, R. and Cho, K. (2019). Passage Re-ranking with BERT. *CoRR*, abs/1901.04085.

[172] Oleynik, M., Faessler, E., Sasso, A. M., Kappattanavar, A., Bergner, B., Cruz, H. F. D., Sachs, J. P., Datta, S., and Böttinger, E. P. (2018). HPI-DHC at TREC 2018 Precision Medicine Track. In *Proc. of the Twenty-Seventh Text REtrieval Conference, TREC 2018, Gaithersburg, Maryland, USA, November 14-16, 2018*, volume 500-331 of *NIST Special Publication*. NIST.

[173] Onal, K. D., Zhang, Y., Altingovde, I. S., Rahman, M. M., Karagoz, P., Braylan, A., Dang, B., Chang, H. L., Kim, H., McNamara, Q., Angert, A., Banner, E., Khetan, V., McDonnell, T., Nguyen, A. T., Xu, D., Wallace, B. C., de Rijke, M., and Lease, M. (2018). Neural information retrieval: at the end of the early years. *Inf. Retr. J.*, 21(2-3):111–182.

[174] Pal, D., Mitra, M., and Datta, K. (2014). Improving Query Expansion Using WordNet. *J. Assoc. Inf. Sci. Technol.*, 65(12):2469–2478.

[175] Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., Song, X., and Ward, R. K. (2014). Semantic Modelling with Long-Short-Term Memory for Information Retrieval. *CoRR*, abs/1412.6629.

[176] Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., Song, X., and Ward, R. K. (2016). Deep Sentence Embedding Using Long Short-Term Memory Networks: Analysis and Application to Information Retrieval. *IEEE ACM Trans. Audio Speech Lang. Process.*, 24(4):694–707.

[177] Pang, L., Lan, Y., Guo, J., Xu, J., and Cheng, X. (2016a). A Study of MatchPyramid Models on Ad-hoc Retrieval. *CoRR*, abs/1606.04648.

[178] Pang, L., Lan, Y., Guo, J., Xu, J., Wan, S., and Cheng, X. (2016b). Text Matching as Image Recognition. In *Proc. of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2793–2799. AAAI Press.

[179] Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global Vectors for Word Representation. In *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar*, pages 1532–1543. ACL.

[180] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep Contextualized Word Representations. In *Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018*, pages 2227–2237. ACL.

[181] Ponte, J. M. and Croft, W. B. (1998). A Language Modeling Approach to Information Retrieval. In *Proc. of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28 1998, Melbourne, Australia*, pages 275–281. ACM.

[182] Purpura, A., Maggipinto, M., Silvello, G., and Susto, G. A. (2019). Probabilistic Word Embeddings in Neural IR: A Promising Model That Does Not Work as Expected (For Now). In *Proc. of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR 2019, Santa Clara, CA, USA, October 2-5, 2019*, pages 3–10. ACM.

[183] Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proc. of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50. ELRA.

[184] Roberts, K., Demner-Fushman, D., Voorhees, E. M., and Hersh, W. R. (2016a). Overview of the TREC 2016 Clinical Decision Support Track. In *Proc. of The Twenty-Fifth Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*. NIST.

[185] Roberts, K., Demner-Fushman, D., Voorhees, E. M., Hersh, W. R., Bedrick, S., and Lazar, A. J. (2018). Overview of the TREC 2018 precision medicine track. In *Proc. of the Twenty-Seventh Text REtrieval Conference, TREC 2018, Gaithersburg, Maryland, USA, November 14-16, 2018*, volume 500-331 of *NIST Special Publication*. NIST.

[186] Roberts, K., Demner-Fushman, D., Voorhees, E. M., Hersh, W. R., Bedrick, S., Lazar, A. J., and Pant, S. (2017). Overview of the TREC 2017 precision medicine track. In *Proc. of The Twenty-Sixth Text REtrieval Conference, TREC 2017, Gaithersburg, Maryland, USA, November 15-17, 2017*, volume 500-324 of *NIST Special Publication*. NIST.

[187] Roberts, K., Demner-Fushman, D., Voorhees, E. M., Hersh, W. R., Bedrick, S., Lazar, A. J., Pant, S., and Meric-Bernstam, F. (2019). Overview of the TREC 2019 precision medicine track. In *Proc. of the Twenty-Eighth Text REtrieval Conference, TREC 2019, Gaithersburg, Maryland, USA, November 13-15, 2019*, volume 1250 of *NIST Special Publication*. NIST.

[188] Roberts, K., Simpson, M., Demner-Fushman, D., Voorhees, E., and Hersh, W. (2016b). State-of-the-art in biomedical literature retrieval for clinical cases: a survey of the TREC 2014 CDS track. *Inf. Retr. Journal*, 19(1-2):113–148.

[189] Roberts, K., Simpson, M. S., Voorhees, E. M., and Hersh, W. R. (2015). Overview of the TREC 2015 Clinical Decision Support Track. In *Proc. of The Twenty-Fourth Text REtrieval Conference, TREC 2015, Gaithersburg, Maryland, USA, November 17-20, 2015*. NIST.

[190] Robertson, S. E. (2004). Understanding Inverse Document Frequency: on Theoretical Arguments for IDF. *Journal of Documentation*, 60(5):503–520.

[191] Robertson, S. E. and Hull, D. A. (2000). The TREC-9 Filtering Track Final Report. In *Proc. of The Ninth Text REtrieval Conference, TREC 2000, Gaithersburg, Maryland, USA, November 13-16, 2000*, volume 500-249 of *NIST Special Publication*. NIST.

[192] Robertson, S. E. and Zaragoza, H. (2009). The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

[193] Rocchio, J. (1965). Relevance Feedback in Information Retrieval. *Report No. ISR-9 to the National Science Foundation, Sect. 23, Computation Laboratory, Harvard University, September, 1965*, pages 1–18.

[194] Rocchio, J. and Salton, G. (1965). Information Search Optimization and Interactive Retrieval Techniques. In *Proc. of the November 30–December 1, 1965, Fall Joint Computer Conference, Part I, Las Vegas, Nevada*, AFIPS '65, pages 293–305. ACM.

[195] Salton, G. (1971). *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Inc., USA.

[196] Salton, G. and McGill, M. J. (1986). *Introduction to modern information retrieval*. McGraw-Hill, Inc.

[197] Sanderson, M. and Croft, W. B. (2012). The History of Information Retrieval Research. *Proceedings of the IEEE*, 100(Centennial-Issue):1444–1451.

[198] Savova, G. K., Masanz, J. J., Ogren, P. V., Zheng, J., Sohn, S., Schuler, K. K., and Chute, C. G. (2010). Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): Architecture, Component Evaluation and Applications. *J. Am. Medical Informatics Assoc.*, 17(5):507–513.

[199] Shah, N. H., Bhatia, N., Jonquet, C., Rubin, D. L., Chiang, A. P., and Musen, M. A. (2009). Comparison of concept recognizers for building the Open Biomedical Annotator. *BMC Bioinformatics*, 10(S-9):14.

[200] Shaw, J. A. and Fox, E. A. (1994). Combination of Multiple Searches. In *Proc. of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*, volume 500-225, pages 105–108. NIST.

[201] Shen, Y., He, X., Gao, J., Deng, L., and Mesnil, G. (2014). A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval. In *Proc. of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, pages 101–110. ACM.

[202] Silvello, G., Bucco, R., Busato, G., Fornari, G., Langeli, A., Purpura, A., Rocco, G., Tezza, A., and Agosti, M. (2018). Statistical Stemmers: A Reproducibility Study. In *Proc. of the 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018*, volume 10772 of *Lecture Notes in Computer Science*, pages 385–397. Springer.

[203] Singhal, A. and Pereira, F. C. N. (1999). Document Expansion for Speech Retrieval. In *Proc. of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '99, August 15-19, 1999, Berkeley, CA, USA*, pages 34–41. ACM.

[204] Sinoara, R. A., Camacho-Collados, J., Rossi, R. G., Navigli, R., and Rezende, S. O. (2019). Knowledge-Enhanced Document Embeddings for Text Classification. *Knowl.-Based Syst.*, 163:955–971.

[205] Sioutos, N., de Coronado, S., Haber, M. W., Hartel, F. W., Shaiu, W. L., and Wright, L. W. (2007). NCI thesaurus: A semantic model integrating cancer-related clinical and molecular information. *J. Biomed. Informatics*, 40(1):30–43.

[206] Soldaini, L., Cohan, A., Yates, A., Goharian, N., and Frieder, O. (2015). Retrieving Medical Literature for Clinical Decision Support. In *Proc. of the 37th European Conference on IR Research, ECIR 2015, Vienna, Austria, March 29 - April 2, 2015*, volume 9022 of *Lecture Notes in Computer Science*, pages 538–549.

[207] Soldaini, L. and Goharian, N. (2016). Quickumls: a fast, unsupervised approach for medical concept extraction. In *MedIR Workshop, SIGIR 2016*.

[208] Soldaini, L., Yates, A., and Goharian, N. (2017). Learning to Reformulate Long Queries for Clinical Decision Support. *J. Assoc. Inf. Sci. Technol.*, 68(11):2602–2619.

[209] Sondhi, P., Sun, J., Zhai, C., Sorrentino, R., and Kohn, M. S. (2012). Leveraging Medical Thesauri and Physician Feedback for Improving Medical Literature Retrieval for Case Queries. *J. Am. Medical Informatics Assoc.*, 19(5):851–858.

[210] Sordoni, A., Bengio, Y., and Nie, J. Y. (2014). Learning Concept Embeddings for Query Expansion by Quantum Entropy Minimization. In *Proc. of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1586–1592. AAAI Press.

[211] Spärck Jones, K. (1972). A Statistical Interpretation of Term Specificity and Its Application in Retrieval. *Journal of Documentation*, 28(1):11–21.

[212] Srinivasan, P. (1996a). Query Expansion and MEDLINE. *Inf. Process. Manag.*, 32(4):431–443.

[213] Srinivasan, P. (1996b). Retrieval Feedback in MEDLINE. *J. Am. Medical Informatics Assoc.*, 3(2):157–167.

[214] Stokes, N., Li, Y., Cavedon, L., and Zobel, J. (2009). Exploring Criteria for Successful Query Expansion in the Genomic Domain. *Inf. Retr.*, 12(1):17–50.

[215] Strohman, T., Metzler, D., Turtle, H., and Croft, W. B. (2005). Indri: A language model-based search engine for complex queries. In *Proc. of the International Conference on Intelligent Analysis*, pages 2–6. Citeseer.

[216] Subramanya, A., Petrov, S., and Pereira, F. C. N. (2010). Efficient Graph-Based Semi-Supervised Learning of Structured Tagging Models. In *Proc. of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9-11 October 2010, MIT Stata Center, Massachusetts, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 167–176. ACL.

[217] Sun, Y., Wang, S., Li, Y., Feng, S., Chen, X., Zhang, H., Tian, X., Zhu, D., Tian, H., and Wu, H. (2019). ERNIE: Enhanced Representation through Knowledge Integration. *CoRR*, abs/1904.09223.

[218] Tague-Sutcliffe, J. (1992). The Pragmatics of Information Retrieval Experimentation Revisited. *Inf. Proc. Manage.*, 28(4):467–490.

[219] Tamborero, D., Rubio-Perez, C., Deu-Pons, J., Schroeder, M. P., Vivancos, A., Rovira, A., Tusquets, I., Albanell, J., Rodon, J., and Tabernero, J. (2018). Cancer Genome Interpreter Annotates the Biological and Clinical Relevance of Tumor Alterations. *Genome Med.*, 10(1):25.

[220] Tamine, L., Soulier, L., Nguyen, G. H., and Souf, N. (2019). Offline Versus Online Representation Learning of Documents Using External Knowledge. *ACM Trans. Inf. Syst.*, 37(4):42:1–42:34.

[221] Thompson, B. (2006). *Foundations of Behavioral Statistics: An Insight-Based Approach*. Guilford Press.

[222] Van Gysel, C., de Rijke, M., and Kanoulas, E. (2016a). Learning Latent Vector Spaces for Product Search. In *Proc. of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, pages 165–174. ACM.

[223] Van Gysel, C., de Rijke, M., and Kanoulas, E. (2018). Neural Vector Spaces for Unsupervised Information Retrieval. *ACM Trans. Inf. Syst.*, 36(4):38:1–38:25.

[224] Van Gysel, C., de Rijke, M., and Worring, M. (2016b). Unsupervised, Efficient and Semantic Expertise Retrieval. In *Proc. of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 1069–1079. ACM.

[225] Van Gysel, C., Kanoulas, E., and de Rijke, M. (2017a). Pyndri: A Python Interface to the Indri Search Engine. In *Proc. of the 39th European Conference on IR Research, ECIR 2017, Aberdeen, UK, April 8-13, 2017*, volume 10193 of *Lecture Notes in Computer Science*, pages 744–748.

[226] Van Gysel, C., Li, D., and Kanoulas, E. (2017b). ILPS at TREC 2017 Common Core Track. In *Proc. of The Twenty-Sixth Text REtrieval Conference, TREC 2017, Gaithersburg, Maryland, USA, November 15-17, 2017*, volume 500-324 of *NIST Special Publication*. National Institute of Standards and Technology (NIST).

[227] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is All you Need. In *Proc. of the 31st Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

[228] Vechtomova, O. (2005). The Role of Multi-word Units in Interactive Information Retrieval. In *Proc. of the 27th European Conference on IR Research, ECIR 2005, Santiago de Compostela, Spain, March 21-23, 2005*, volume 3408 of *Lecture Notes in Computer Science*, pages 403–420. Springer.

[229] Voorhees, E. M. (1993). Using WordNet to Disambiguate Word Senses for Text Retrieval. In *Proc. of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Pittsburgh, PA, USA, June 27 - July 1, 1993*, pages 171–180. ACM.

[230] Voorhees, E. M. (1994). Query Expansion Using Lexical-Semantic Relations. In *Proc. of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland, 3-6 July 1994*, pages 61–69. ACM.

[231] Voorhees, E. M. (2001). Evaluation by Highly Relevant Documents. In *Proc. of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, September 9-13, 2001, New Orleans, Louisiana, USA*, pages 74–82. ACM.

[232] Voorhees, E. M. (2005). The TREC robust retrieval track. *SIGIR Forum*, 39(1):11–20.

[233] Voorhees, E. M. and Harman, D. K. (2005). *TREC: Experiment and Evaluation in Information Retrieval*, volume 63. The MIT Press.

[234] Voorhees, E. M. and Tong, R. M. (2011). Overview of the TREC 2011 Medical Records Track. In *Proc. of The Twentieth Text REtrieval Conference, TREC 2011, Gaithersburg, Maryland, USA, November 15-18, 2011*, NIST Special Publication. NIST.

[235] Vulić, I. and Moens, M. F. (2015). Monolingual and Cross-Lingual Information Retrieval Models Based on (Bilingual) Word Embeddings. In *Proc. of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, pages 363–372. ACM.

[236] Wan, S., Lan, Y., Xu, J., Guo, J., Pang, L., and Cheng, X. (2016). Match-SRNN: Modeling the Recursive Matching Structure with Spatial RNN. In *Proc. of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2922–2928. IJCAI/AAAI Press.

[237] Wand, M. P. and Jones, M. C. (1995). *Kernel Smoothing*. Springer.

[238] Wei, X. and Croft, W. B. (2006). LDA-Based Document Models for Ad-Hoc Retrieval. In *Proc. of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006*, pages 178–185. ACM.

[239] Xiong, C., Dai, Z., Callan, J., Liu, Z., and Power, R. (2017). End-to-End Neural Ad-hoc Ranking with Kernel Pooling. In *Proc. of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 55–64. ACM.

[240] Xu, C., Bai, Y., Bian, J., Gao, B., Wang, G., Liu, X., and Liu, T. Y. (2014). RC-NET: A General Framework for Incorporating Knowledge into Word Representations. In *Proc. of the 23rd ACM International Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, pages 1219–1228. ACM.

[241] Xu, J. and Croft, W. B. (2017). Query Expansion Using Local and Global Document Analysis. *SIGIR Forum*, 51(2):168–175.

[242] Yamada, I., Shindo, H., Takeda, H., and Takefuji, Y. (2016). Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation. In *Proc. of the 20th Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 250–259. ACL.

[243] Yang, P., Fang, H., and Lin, J. (2018a). Anserini: Reproducible ranking baselines using lucene. *J. Data and Information Quality*, 10(4):16:1–16:20.

[244] Yang, P., Fang, H., and Lin, J. (2018b). Anserini: Reproducible Ranking Baselines Using Lucene. *J. Data and Information Quality*, 10(4):16:1–16:20.

[245] Yang, W., Lu, K., Yang, P., and Lin, J. (2019a). Critically Examining the "Neural Hype": Weak Baselines and the Additivity of Effectiveness Gains from Neural Ranking Models. In *Proc. of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 1129–1132. ACM.

[246] Yang, W., Zhang, H., and Lin, J. (2019b). Simple Applications of BERT for Ad Hoc Document Retrieval. *CoRR*, abs/1903.10972.

[247] Yang, X., Ounis, I., McCreadie, R., Macdonald, C., and Fang, A. (2018c). On the Reproducibility and Generalisation of the Linear Transformation of Word Embeddings. In *Proc. of the 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018*, volume 10772 of *Lecture Notes in Computer Science*, pages 263–275. Springer.

[248] Ye, X., Qi, Z., and Massey, D. (2015). Learning Relevance from Click Data via Neural Network Based Similarity Models. In *Proc. of the 2015 IEEE International Conference on Big Data, Big Data 2015, Santa Clara, CA, USA, October 29 - November 1, 2015*, pages 801–806. IEEE.

[249] Yilmaz, E., Kanoulas, E., and Aslam, J. A. (2008). A simple and efficient sampling method for estimating AP and NDCG. In *Proc. of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008*, pages 603–610. ACM.

[250] Yu, M. and Dredze, M. (2014). Improving Lexical Embeddings with Semantic Knowledge. In *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA*, pages 545–550. ACL.

[251] Zamani, H. and Croft, W. B. (2016a). Embedding-based Query Language Models. In *Proc. of the 2016 ACM International Conference on the Theory of Information Retrieval, ICTIR 2016, Newark, DE, USA, September 12- 6, 2016*, pages 147–156. ACM.

[252] Zamani, H. and Croft, W. B. (2016b). Estimating Embedding Vectors for Queries. In *Proc. of the 2016 ACM International Conference on the Theory of Information Retrieval, ICTIR 2016, Newark, DE, USA, September 12- 6, 2016*, pages 123–132. ACM.

[253] Zargayouna, H., Roussey, C., and Chevallet, J. P. (2015). Recherche d'information sémantique : état des lieux. *TAL*, 56(3).

[254] Zhai, C. and Lafferty, J. D. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214.

[255] Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., and Liu, Q. (2019). ERNIE: Enhanced Language Representation with Informative Entities. In *Proc. of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019*, pages 1441–1451. ACL.

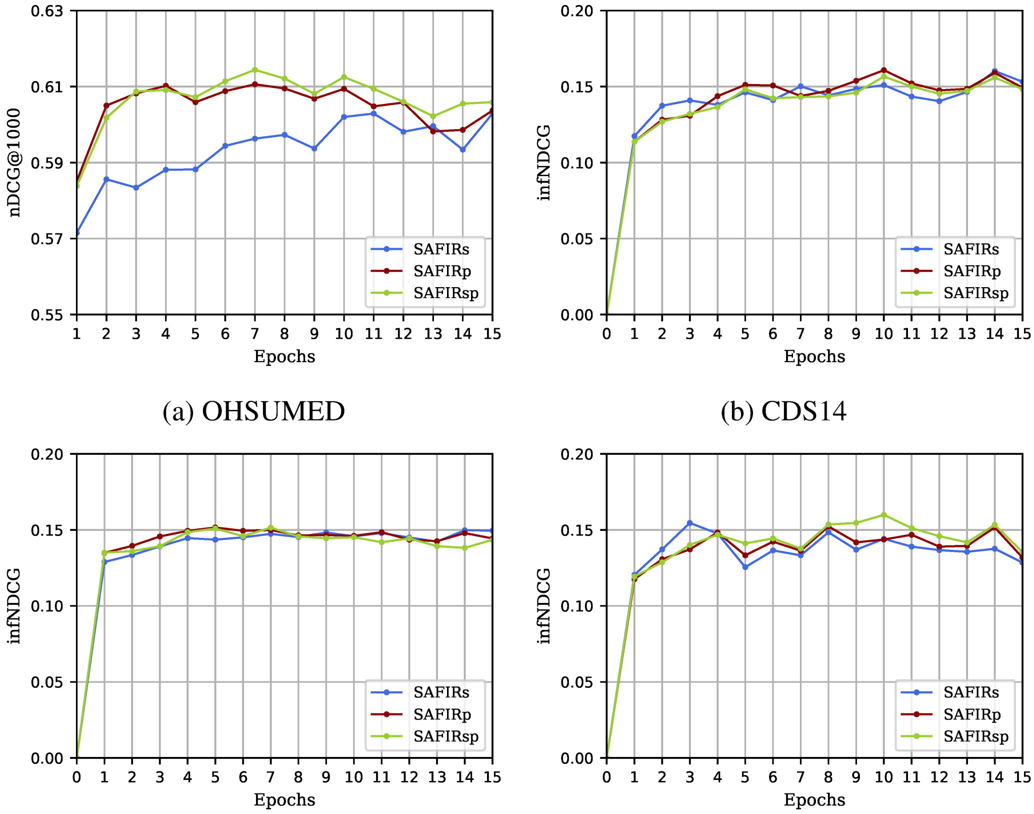
[256] Zhu, D., Wu, S. T., Carterette, B., and Liu, H. (2014). Using Large Clinical Corpora for Query Expansion in Text-Based Cohort Identification. *J. Biomed. Informatics*, 49:275–281.

[257] Zuccon, G., Koopman, B., Bruza, P., and Azzopardi, L. (2015). Integrating and Evaluating Neural Word Embeddings in Information Retrieval. In *Proc. of the 20th Australasian Document Computing Symposium, ADCS 2015, Parramatta, NSW, Australia, December 8-9, 2015*, pages 12:1–12:8. ACM.

Appendix A

SAFIR Variants Averaged Performances

The behavior of SAFIR variants in terms of optimization as training progresses is shown in Figure A.1.



(a) OHSUMED

(b) CDS14

(c) CDS15

(d) CDS16

Fig. A.1 nDCG@1000/infNDCG scores as training of SAFIR variants progresses on each collection. For OHSUMED, we present the optimization results from epoch 1 to ease visualization.

The curves show that SAFIR variants improve up to a certain value and then start to oscillate across iterations – although these oscillations tend to be small in most cases. Therefore, we investigate how the performances change when we consider the average over iterations 10-15 instead of the best iteration. To this end, we compare SAFIR variants with NVSM – averaged over iterations 10-15 – and BM25/RM3. The results are reported in Table A.1.

Table A.1 Retrieval performances of considered models averaged over epochs 10-15. Models are grouped by type: Bag-of-Words (BoW), Corpus-Driven (CD), and SAFIR. The values in parentheses represent the standard deviation values. **Bold** values represent the highest scores among the models in each collection.

		infNDCG				nDCG@1000			
		CDS14	CDS15	CDS16	OHSUMED	CDS14	CDS15	CDS16	OHSUMED
BoW	BM25/RM3	0.1384 (0.0000)	**0.1578** (0.0000)	**0.1688** (0.0000)	– (–)	0.2316 (0.0000)	0.2183 (0.0000)	**0.2068** (0.0000)	**0.6253** (0.0000)
CD	NVSM	0.1507 (0.0061)	0.1436 (0.0012)	0.1367 (0.0045)	– (–)	0.2658 (0.0030)	0.2168 (0.0061)	0.1839 (0.0037)	0.5947 (0.0021)
SAFIR	SAFIR$_s$	0.1491 (0.0066)	0.1467 (0.0026)	0.1369 (0.0046)	– (–)	0.2559 (0.0031)	0.2222 (0.0021)	0.1778 (0.0039)	0.6002 (0.0040)
	SAFIR$_p$	**0.1529** (0.0052)	0.1455 (0.0022)	0.1421 (0.0063)	– (–)	**0.2696** (0.0032)	**0.2242** (0.0020)	0.1882 (0.0025)	0.6034 (0.0040)
	SAFIR$_{sp}$	0.1506 (0.0043)	0.1421 (0.0026)	0.1479 (0.0080)	– (–)	0.2652 (0.0032)	0.2195 (0.0019)	0.1832 (0.0025)	0.6069 (0.0033)

		nDCG@100				nDCG@10			
		CDS14	CDS15	CDS16	OHSUMED	CDS14	CDS15	CDS16	OHSUMED
BoW	BM25/RM3	0.1338 (0.0000)	**0.1522** (0.0000)	**0.1298** (0.0000)	**0.4746** (0.0000)	0.1645 (0.0000)	**0.1986** (0.0000)	**0.1518** (0.0000)	**0.4618** (0.0000)
CD	NVSM	0.1347 (0.0042)	0.1348 (0.0024)	0.1033 (0.0040)	0.4140 (0.0028)	0.1611 (0.0147)	0.1636 (0.0030)	0.1236 (0.0094)	0.3793 (0.0062)
SAFIR	SAFIR$_s$	0.1340 (0.0040)	0.1386 (0.0026)	0.1029 (0.0028)	0.4196 (0.0055)	0.1639 (0.0108)	0.1726 (0.0098)	0.1218 (0.0112)	0.4040 (0.0068)
	SAFIR$_p$	**0.1420** (0.0040)	0.1378 (0.0017)	0.1077 (0.0042)	0.4254 (0.0049)	0.1769 (0.0121)	0.1821 (0.0024)	0.1450 (0.0146)	0.4038 (0.0042)
	SAFIR$_{sp}$	0.1412 (0.0031)	0.1365 (0.0020)	0.1083 (0.0034)	0.4308 (0.0042)	**0.1869** (0.0109)	0.1694 (0.0046)	0.1493 (0.0121)	0.4183 (0.0088)

		P@10				Recall@1000			
		CDS14	CDS15	CDS16	OHSUMED	CDS14	CDS15	CDS16	OHSUMED
BoW	BM25/RM3	0.1833 (0.0000)	**0.2433** (0.0000)	**0.2067** (0.0000)	**0.5413** (0.0000)	0.3151 (0.0000)	0.2884 (0.0000)	**0.3059** (0.0000)	0.8431 (0.0000)
CD	NVSM	0.1933 (0.0176)	0.2274 (0.0063)	0.1439 (0.0068)	0.4376 (0.0079)	**0.3903** (0.0036)	0.3073 (0.0120)	0.2765 (0.0032)	**0.8582** (0.0016)
SAFIR	SAFIR$_s$	0.1922 (0.0101)	0.2272 (0.0073)	0.1328 (0.0139)	0.4598 (0.0066)	0.3703 (0.0052)	**0.3123** (0.0024)	0.2661 (0.0060)	0.8564 (0.0018)
	SAFIR$_p$	0.2161 (0.0124)	0.2350 (0.0069)	0.1644 (0.0133)	0.4659 (0.0067)	0.3839 (0.0032)	0.3107 (0.0047)	0.2794 (0.0051)	0.8564 (0.0030)
	SAFIR$_{sp}$	**0.2195** (0.0097)	0.2211 (0.0097)	0.1656 (0.0170)	0.4733 (0.0120)	0.3723 (0.0030)	0.3053 (0.0031)	0.2697 (0.0026)	0.8504 (0.0019)

The results from Table A.1 show that the performances obtained by SAFIR variants averaged over iterations 10-15 are similar – although often lower – to those obtained with the best iterations (cf. Table 6.11). The most notable exceptions are nDCG@10 and P@10, where the differences between averaged and best performances are larger. However, precision-oriented measures are highly sensitive to performance variations. Therefore, the different representations used at each iteration to perform retrieval can have a large impact on the results for these measures – especially when the considered representations are far from being optimal.

Overall, the models that perform best are consistent between averaged and best iterations. In particular, the average of SAFIR over iterations 10-15 achieves top performances in most of the measures in which also the best iteration of SAFIR achieves them. The only exceptions are P@10 in CDS15 and Recall@1000 in CDS14, where BM25/RM3 and NVSM achieve the best results, respectively. However, the SAFIR variants that achieve top results are different for some measures in CDS collections. SAFIR_{sp} achieves top performances instead of SAFIR_{p} for nDCG@10 and P@10 in CDS14, whereas SAFIR_{p} replaces SAFIR_{sp} for nDCG@1000 in CDS15. On the other hand, the ranking of the models for OHSUMED does not change regardless of the approach selected – be it the average of the iterations 10-15 or the best iteration. The only exception is for nDCG@10, where SAFIR_{s} outperforms SAFIR_{p}.

To understand to what extent the rankings of considered models change when we take the average of iterations 10-15 instead of the best iteration, we perform Kendall's τ correlations between model rankings obtained in one way or the other. Table A.2 reports correlation values.

Table A.2 Kendall τ correlations computed between the rankings of the considered models from Table A.1 (average of iterations 10-15) and Table 6.11 (best iteration) for each measure in each collection.

	infNDCG	nDCG@1000	nDCG@100	nDCG@10	P@10	Recall@1000
OHSUMED	–	1.00	1.00	0.80	1.00	1.00
CDS14	0.60	0.80	0.80	0.40	0.80	0.80
CDS15	0.40	0.40	0.80	0.60	0.20	0.60
CDS16	0.80	0.80	0.80	0.40	0.20	0.80

Table A.2 shows that in more than 60% of cases correlation values are greater than or equal to 0.80 – which indicates that the differences between rankings do not reflect noticeable changes [231]. The rest of the correlation values divides among 0.60 (13% of cases), 0.40 (17% of cases), and 0.20 (9% of cases). Given the short length of the considered ranking lists (only five elements), a correlation value of 0.80 means that the two rankings differ by a single swap between positions. As a result, correlation values of 0.60 occur with two swaps between positions of the ranking list, whereas scores of 0.40 and 0.20 with three and four swaps, respectively. Below, we focus on low correlation values – i.e., 0.40 and 0.20 – and we detail the differences between the ranking lists obtained taking the average of iterations 10-15 and the best iteration.

As expected, low correlations cluster on precision-oriented measures. nDCG@10 presents correlation values of 0.40 in CDS14 and CDS16. In CDS14, SAFIR$_{\mathrm{sp}}$ outperforms SAFIR$_{\mathrm{p}}$ and becomes the top performing model, whereas BM25/RM3 moves from last to the third position. In CDS16, BM25/RM3 and SAFIR$_{\mathrm{sp}}$ outperform SAFIR$_{\mathrm{p}}$ achieving the first and second positions, respectively. On the other hand, SAFIR$_{\mathrm{s}}$ moves from the fourth position to the last. As for P@10, CDS15 and CDS16 show correlation values of 0.20. In CDS15, BM25/RM3 moves from the third to the top position. Also, SAFIR$_{\mathrm{s}}$ outperforms SAFIR$_{\mathrm{sp}}$ – which becomes the worst performing model. In CDS16, SAFIR$_{\mathrm{sp}}$ outperforms both SAFIR$_{\mathrm{s}}$ and SAFIR$_{\mathrm{p}}$ achieving the second position, whereas SAFIR$_{\mathrm{s}}$ moves to the last position.

Other than nDCG@10 and P@10, CDS15 exhibits low correlations (0.40) also for infNDCG and nDCG@1000. For infNDCG, SAFIR$_{\mathrm{s}}$ gains two positions (from fourth to second) and SAFIR$_{\mathrm{sp}}$ becomes the worst performing model. For nDCG@1000, SAFIR$_{\mathrm{p}}$ and SAFIR$_{\mathrm{s}}$ outperform SAFIR$_{\mathrm{sp}}$ achieving the first and second positions, respectively. Moreover, BM25/RM3 outperforms NVSM and moves from the last to the fourth position.

Thus, the results of this analysis show that the performances obtained by SAFIR variants averaged over iterations 10-15 are similar – although often lower – to those obtained with the best iterations. Furthermore, the average of SAFIR over iterations 10-15 achieves top performances in most of the measures in which also the best iteration of SAFIR achieves them. However, the SAFIR variants that achieve top results are different for some measures. Finally, the rankings obtained when we take the average of iterations 10-15 present high correlation values with the rankings obtained considering the best iteration in most cases. As expected, most of the low correlations occur for precision-oriented measures – which are highly sensitive to variations in models performance.