FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Word Representation for Text Analysis and Search

Document Retrieval, Sentiment Analysis, and Cross Lingual Word Sense Disambiguation

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der Technischen Wissenschaften

by

Dipl.-Ing. Navid Rekabsaz, BSc
Registration Number 1129057

to the Faculty of Informatics

at the TU Wien

Advisor: Prof. Dr. Allan Hanbury
Co-advisor: Mihai Lupu, PhD

The dissertation has been reviewed by:

Carsten Eickhoff

Christina Lioma

Vienna, 4th April, 2018

Navid Rekabsaz

Erklärung zur Verfassung der Arbeit

Dipl.-Ing. Navid Rekabsaz, BSc
Favoritenstrasse 9 HD 01 07, 1040 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 4 April, 2018

Navid Rekabsaz

Acknowledgements

I would like to express my sincere appreciation of my excellent supervisors Professor Allan Hanbury and Dr. Mihai Lupu. Their outstanding qualities, which include openness, patience, generosity of support, and a great sense of responsibility, are indeed the pillars on which I could complete this thesis successfully.

With all love, I devote this thesis to Sophia, for all the lights and beauties that she is, and has brought to my life.

Kurzfassung

Semantik in der Sprache ist ein grundlegender Aspekt der menschlichen Erkenntnis und bestimmt in hohem Maße unser Verständnis und Wissen. Methoden der Wort-Repräsentation bringen ein Computermodell ein zur Erfassung von Semantik, in dem Vektoren als Proxys für die Bedeutung von Begriffen bereitgestellt werden, was als "Wort-Embedding" bekannt ist. Neueste Weiterentwicklungen des Modells durch den Einsatz neuronaler Netzwerke eröffnen eine spannende Perspektive und drängen auf weitere Forschung zum Verständnis und zur Verwendung semantischer Repräsentations-Modelle in der Sprach- und Textverarbeitung.

In dieser Dissertation stellen wir neue Methoden vor für die Nutzung von Wort-Repräsentations Modellen in verschiedenen Textanalyse Aufgabenstellungen. Wir bieten auch eingehende Analysen des Konzepts der Begriffs-Verwandtschaft in semantischen Modellen. Die Arbeit leistet einen Beitrag zur Grundlagenforschung auf dem Gebiet des Informations-Retrieval und der Interpretierbarkeit von Wort-Repräsentationen, sowie zur angewandten Forschung in der sprachübergreifenden Wortbedeutungs-Disambiguierung (CL-WSD) und der Sentiment-Analyse. Wir befassen uns mit verschiedenen Aufgaben des Informationsmanagements, wie des Dokumenten-Retrieval, der Gender-Bias-Erkennung, der CL-WSD für Sprachen, für die kaum Ressourcen vorhanden sind, und mit Volatilitätsprognosen, die in den Bereichen Nachrichten, Gesundheit, Finanzen, und Sozialwissenschaften erstellt werden.

In der ersten Aufgabe—des Dokumenten-Retrieval—führen wir einen neuartigen Ansatz ein, um die Informationen, die von verwandten Begriffen gewonnen werden, in traditionelle Retrieval-Modelle zu integrieren. Der Ansatz verallgemeinert die Idee der Translations-Modelle für verschiedene probabilistische Modelle. Im Verlauf der Studie erkennen wir, wie wichtig es dabei ist, zwei relevante Themen zu beachten: wie man die verwandten Begriffe in den Repräsentations-Modellen auswählt, und wie man Ähnlichkeiten zwischen Begriffen an die spezifischen Bedürfnisse von Retrieval-Systemen anpasst. Wir nähern uns ersterem Thema, indem wir den Raum von Wort-Vektoren untersuchen, und letzterem, indem wir Ähnlichkeiten von Repräsentationen kombinieren, die auf verschiedenen Annahmen über die umgebenden Begriffs-Kontexte basieren. Unsere Evaluierung mehrerer Retrieval-Test-Sammlungen zeigt signifikante Verbesserungen in der Suchleistung durch die Anwendung der verallgemeinerten Translations-Modelle gegenüber starker Ausgangswerte auf dem letzten Stand der Technik.

Das nächste Thema befasst sich mit der Interpretierbarkeit des Wort-Embedding mittels Einführung eines neuartigen neuronalen Repräsentations-Modells. Das Modell überträgt dichtes Wort-Embedding auf dünn besetzte Vektoren, für die die semantischen Konzepte der Repräsentationen explizit bestimmt sind. Als Fallstudie verwenden wir diese expliziten Repräsentationen, um den Grad von Gender-Bias in Wikipedia-Artikeln zu quantifizieren. Unsere Analyse zeigt starke Verzerrungen in einigen spezifischen Berufen (z.B. “nurse”) in Richtung weibliche Konnotation.

Die nächste Aufgabe betrifft CL-WSD für ressourcenarme Sprachen / Domänen (von Englisch zu Persisch in unserer Arbeit). Wir nähern uns dieser Aufgabe mittels der semantischen Ähnlichkeit von übersetzten Begriffen in ihren jeweiligen Kontexten und zeigen die Vorteile der Nutzung von Wort-Repräsentationen für CL-WSD, insbesondere in Abwesenheit von zuverlässigen Ressourcen.

Schließlich tragen wir zum letzten Stand der Technik in der Sentiment-Analyse bei, indem wir die verallgemeinerten Translations-Modelle zur Vorhersage der Volatilität an Finanzmärkten nutzen. In Kombination mit tatsächlichen Marktdaten übertrifft unser Ansatz andere State-of-the-Art Methoden und zeigt die Vorteile der Verwendung von textuellen Daten zusammen mit semantischen Methoden für Volatilitätsprognosen.

Abstract

Semantics in language is a fundamental aspect of human cognition and in great extent defines our understanding and knowledge. Word representation methods suggest a computational model to capture semantics by providing vectors as proxies to the meaning of terms, known as word embedding. Recent advancements of the models using neural network approaches open an exciting perspective, and urge further research on understanding and making use of semantic representation models in language and text processing.

In this thesis, we introduce novel methodologies to exploit word representation models in various text analysis tasks. We also provide in-depth analyses of the concept of term relatedness in semantic models. The thesis contributes to basic research in the area of Information Retrieval and word representation interpretability, as well as applied research in Cross-Lingual Word Sense Disambiguation (CL-WSD), and sentiment analysis. We cover several tasks in Information Management such as document retrieval, gender bias detection, CL-WSD for language with scarce resources, and volatility prediction, studied in the news, health, finance, and social science domains.

In the first task—document retrieval—we introduce a novel approach to integrate the information of related terms in traditional retrieval models. The approach generalizes the idea of the translation model to various probabilistic models. In the course of the study, we realize the importance of addressing two relevant topics: how to select the related terms in the representation models, and how to adapt the term similarities to the specific needs of retrieval systems. We approach the former by exploring the space of word vectors, and the latter by combining similarities of representations, created based on different assumptions on the surrounding contexts of terms. Our evaluations on various retrieval test collections show significant improvements in search performance by using the generalized translation models in comparison to strong, state of the art baselines.

The next topic approaches the interpretability of word embedding by introducing a novel neural-based representation model. The model transfers dense word embedding to sparse vectors where the semantic concepts of the representations are explicitly specified. As a case-study, we use these explicit representations to quantify the degree of the existence of gender bias in the Wikipedia articles. Our analysis shows strong bias in a few specific occupations (e.g. nurse) to female.

The next task regards CL-WSD for low-resource languages/domains (English to Persian in our work). We approach this task using the semantic similarity of the translation terms in their contexts, showing the benefits of exploiting word representation for CL-WSD, specially in the absence of reliable resources.

Finally, we contribute to the state-of-the-art of sentiment analysis, by exploiting the generalized translation models to predict volatility in financial markets. Our approach, when combined with factual market data, outperforms state-of-the-art methods, and shows the advantages of using textual data together with semantic methods for volatility forecasting.

Contents

CHAPTER 1

Introduction

Language, to a great extent, defines who we are, by forming our fundamental abilities: thinking, reflecting, communicating, and knowing. Analyzing and understanding text, a resource encompassing subtleties of human language, is a fascinating yet intricate challenge in Artificial Intelligence (AI). The complexity of language makes it difficult to clearly deal with the notion of *language understanding* in AI and indeed makes it an intriguing topic for research and contemplating.

An essential building block of understanding language is the comprehension of the underlying meaning of words—*semantics*—and of the relations/similarities between words—*relatedness*. Studying the semantics of words has been the concern of many linguists, philosophers, and thinkers throughout history. In the last decades, the *computational semantics* field brought this concept to the computer science world. In general, resources in computational semantics are created based on two main approaches: *knowledge* annotation and *statistical* computation.

The knowledge-based approaches mainly rely on encoding the knowledge of experts in lexical resources. These resources usually contain definitions of terms as well as their relations, represented in data structures such as graphs, hierarchies, and sets (e.g. synsets). WordNet [Fel98], BabelNet [NP10], and Dbpedia [ABK+07] are some publicly available examples of such resources. While these knowledge resources provide a valuable and (fairly) accurate representation of language, creating and expanding them is highly expensive and time consuming. This obstacle makes it hard for these resources to cover a wide range of concepts/words of a language (lack of completeness), adapt to the changes in languages or the emergence of new concepts, and to be extended to new languages and domains.

As an alternative approach, statistical semantics suggests a data-oriented solution that relies on finding patterns of term occurrences in large amounts of text data. Such semantic models can be easily created from the text in any language or domain with much less

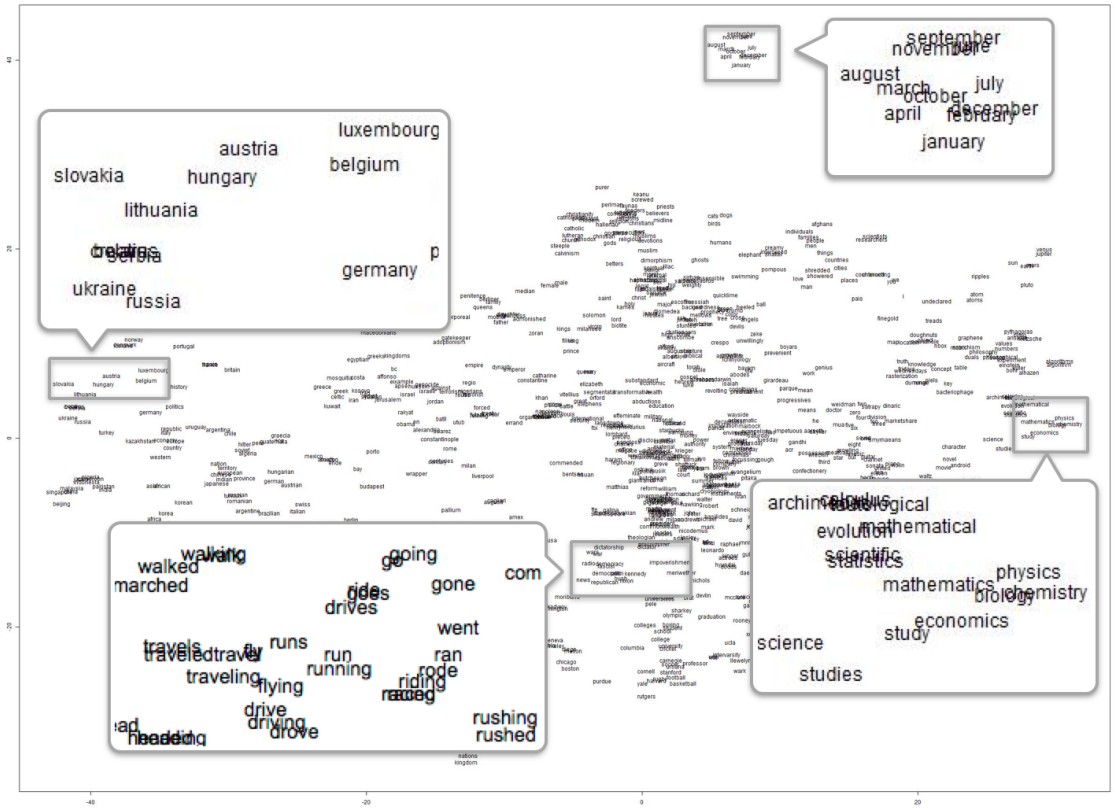


Figure 1.1: A sample semantic representation of a limited number of terms, projected into two-dimensional space.

human resources than knowledge-based approaches. The words in statistical approaches are commonly represented by real-valued high dimensional vectors, referred to as *semantic representations of words*. In these vectors, the dimensions stand for some explicit or implicit concepts in language. In addition to the term "semantic representations of words", several similar names such as "semantic vectors", "word representations", and "word embedding" are widely used in literature. Although these names can refer to slightly different representations, in this thesis we used them interchangeably unless the differences are explicitly explained. Also, as the focus of the thesis is on statistical methods, in general we simply use the term "semantics" instead of "statistical semantics".

The methods to create a semantic representation generally follow one core idea: terms that share common contexts (terms surrounding a term) should have similar vector representations and therefore be semantically related. These methods read the contexts of the terms in a corpus and eventually represent each term by a vector such that the semantically related terms are (geometrically) close to each other in the corresponding high-dimensional space. A sample semantic representation for a small set of terms is shown in Figure 1.1. For the sake of visualization, the original vectors are projected into

a two-dimensional space. As shown, terms with some semantic relations (e.g. months, countries, verbs about moving, etc.) are close to each other.

Semantic vectors have been widely used in various text analysis applications. The main application domain in this thesis is *Information Retrieval (IR)*, "a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information" [Sal68]. Search engines are the most well-known products of this field. In a typical IR scenario, a user queries information and an IR system retrieves the most relevant documents from a data collection. IR is an exciting and challenging domain and can benefit from the information provided by semantics methods. As discussed throughout the thesis, the relations between terms, provided by semantic models, can be effectively exploited for search performance.

In addition to IR, this thesis contributes to the state-of-the-art in the *Natural Language Processing (NLP)* community. The NLP domain deals with the challenges of analyzing, interpreting, and understanding complexity of human language through computer programs. Understanding semantics and especially the semantics of words plays an important role in research in NLP. The studies in this work contribute to NLP topics such as sentiment analysis, semantic vector interpretability, and cross-lingual word sense disambiguation.

The rest of the chapter is organized as follows: In Section 1.1, we review the philosophical ideas, grounding the basis of modern statistical semantics. Section 1.2 points out the open challenges and describes the motivations of this work. In Section 1.3, we explain in detail the contributions of the work, followed by Section 1.4 to plot the structure and road-map of the thesis.

1.1 Historical Paradigms in AI and Language Processing

The history of research in AI, since its birth in 1950s, has experienced three consecutive eras, each following one of the two main philosophical paradigms of knowledge acquisition: *Empiricism* and *Rationalism*. According to *A Dictionary of Philosophy* [Lac96], Empiricism stands for the idea that "knowledge is distilled from one's experiences", while Rationalism refers to "any view appealing to reason as a source of knowledge or justification".

In this section, we first review these two paradigms in the perspective of AI history and language processing—a central research topic of AI. We then briefly study the linguistic theories on (empirical) computational semantics, followed by a discussion on the advantages as well as limitations of such approaches. This section considerably owes its ideas to the following articles: [Wil08, Wil11, Chu11, Ste11].

1.1.1 Rationalism vs. Empiricism

The first era of AI research (1950s–1970s) followed the ideas of Empiricism through data-oriented approaches. The Empiricism paradigm in AI relies on the exploration of

knowledge in existing data to acquire understanding, to predict the behavior of a system, or to appropriately react in a situation. Some of the outstanding works of this era are Shannon's information theory [Sha48], Harris advocations on the close relation between grammatical analysis of natural language and information-theoretic principles [Har51], and Firth positions on context-dependent nature of semantics [Fir57].

In the early 1970's, the interest in Empiricism faded through significant criticisms by rationalist positions, specially in the work of Chomsky [Cho57], and Minsky and Papert [MP69]. The Rationalism paradigm proposes the explicit definition of knowledge using a set of defined rules, sometimes referred to as *rule-based* or *symbolic* AI. Approaches in this paradigm rely on explicit knowledge definition from human experts. The knowledge of a particular domain is encapsulated in a knowledge resource, used by an AI agent to find an answer for a problem/question via applying the set of (pre-defined) rules.

As an example in the area of language processing, the work of Chomsky on formal linguistics [Cho57] explains language by means of a set of rules, defined through representation tools such as automata with formations and transformations. The task of such rule-based representations is to separate meaningful from meaningless expressions. A fundamental assumption in this paradigm is that there exists a reliable and general syntactic well-formedness in every language, based on which we can fully define a language if we only discover and then formalize all the rules.

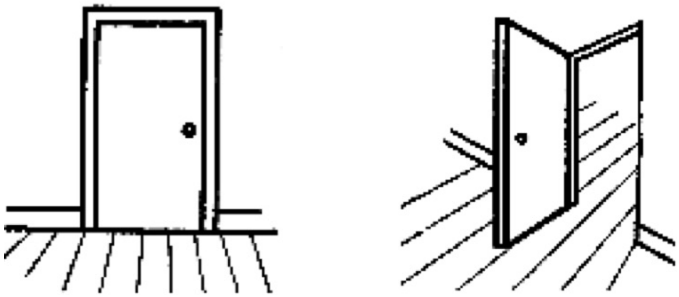
During this period (1970s–1990s), the main thrust of work in NLP was in the search for local and deep, grammatical relations in English but with little concern about words themselves or their effects in language. The hand-coded lexicographies are some of the valuable resources of these lines of work, still actively used in research communities.

The third era of AI research, started in the late 1980s and continues to the present day, witnesses the revival of Empiricism. Specially in the last years due to the exponential increase of data, the data-oriented paradigm has been particularly attracting more attention, and is the main direction of this thesis.

1.1.2 Computational Semantics: An Empirical Approach

Before discussing the computational approaches to semantics, let us first gain a better understanding of the notion of "semantics", a concept which has been extensively studied in various disciplines such as philosophy, linguistics, and psychology. A well-known approach to understand *what is semantics* is by the use of stick-picture situations; a method used for decades to teach a new language. The stick-pictures method expresses a simple proposition in an unambiguous situation. An example of a stick-picture is shown in Figure 1.2.

The data-oriented algorithms for capturing the notion of semantics generally owe their core ideas to the work of Firth [Fir57], summarized with the memorable line: "You shall know a word by the company it keeps." The spirit of this idea is represented in the computational semantics field with a view in which words take on meaning from



(a) The door is shut. (b) The door is open.

Figure 1.2: An example of defining semantics with stick-picture situations. Extracted from [Wil08]

their neighbors (the words that co-occur in a context around a word). Wilks [Wil08] associates the Empiricism paradigm in computational linguistics to even before the "birth of AI", to Wittgenstein's later work on language, the Philosophical Investigations [Wit53]. Wittgenstein views the meaning of words in regard to their usage in language by turning the attention to the activity of using language. Wittgenstein encapsulates this idea into the notion of language games. The idea of language games reflects an analogy between the rules of language and the rules of games, as if saying a sentence or proposition in a language is like making a move in a game. Wittgenstein uses this analogy to demonstrate that the meaning of words depends on their usage in the various and multiform activities of human. Wittgenstein, of course, did not know about computers in the modern form. However, the core of his thesis on the nature of language resonates with the modern data-oriented approaches in computational linguistics.

The Empiricism in computational semantics, interestingly, shares many principal assumptions with IR: a view that language consists only of words with no meta-codings of concepts, and understanding the meaning of words is embedded in their usage in the language corpus. The rejection of such meta-coding in IR is summarized by Spark Jones [Jon03]: "taking words as they stand". This view of language closely resonates with our approaches to IR challenges in this thesis.

Despite the promising advancements and perspectives provided by data-oriented methods to semantics, these approaches also inherit some common issues or inadequacies, mentioned in the following.

The first challenge is that as the word representation methods fundamentally consider the related words as the words with similar contexts, they generalize all different representation relations e.g. antonyms, hypernyms, hyponyms, co-hyponyms, synonyms, antonyms, etc. into one notion of relatedness. This lack of information can be insufficient in many application e.g. in IR, using all these types can bias the search to unrelated topics (e.g. searching for "dog" instead of "cat"!). Kruzewski and Baroni [KB15] notice this by an example: *"[in a statistical semantic model,] 'animal', 'puppy', and 'cat' are all closely*

related to 'dog', but if you tell me that Fido is a dog, I will conclude that he is an animal, that he is not a cat, and that he might or might not be a puppy."

The second issue of the semantic word representation methods is that they and their corresponding mathematical functions can provide an approximation on the relatedness of any two words, although this relatedness could be perceived as completely meaningless in the language. Karlgren et al. [KHS08] point it out by examples, showing that word representation methods are too ready to provide answers to meaningless questions: *"What is more similar to a computer: a sparrow or a star?"*, or *"Is a cell more similar to a phone than a bird is to a compiler?"*

This last challenge concerns the fundamental characteristics of any statistical method: the more frequent incidents define the main characteristics of the model. Wilks [Wil08] describes it with an example: *"Imagine asking the audience in a quiz show 'Who wants to become a Millionaire?'. The most frequent answer is usually [assumed to be correct], but not always [necessarily], correct."* This trait is also present in the algorithms that model semantic representations: in the semantic space, more frequent words tend to become semantically similar with much more other words in comparison with less frequent ones, which is due to the existence of more information (contexts) for the frequent words and not necessarily their intrinsic meanings.

These issues are the natural challenges of empirical approaches.

1.2 Motivations and Research Questions

As mentioned, an established method for quantifying the relatedness between words is the use of semantic word representations or word embeddings. These representation vectors are proxies of the meaning of words and distance functions are proxies of semantic relatedness. Fundamentally, word embedding models exploit the contextual information of the target words in a corpus to approximate their vectors, and hence their relations to other words.

It is indeed an exciting time to study word representation learning and its applications on text analysis tasks. On one hand the extensive amount of available data as well as computational resources and on the other hand the advancements in algorithms make research on representation learning and statistical semantics highly interesting. In particular, recent developments on word embeddings based on neural networks principles provide a novel source of information for term-to-term relatedness.

A longstanding research question in IR is the problem of introducing semantics into document retrieval. Semantic vectors became popular with Latent Semantic Analysis/Indexing (LSA/LSI) [DDF+90] in the early 1990s. Probabilistic Latent Semantic Indexing [Hof99] (pLSI), Latent Dirichlet Allocation [BNJ03] (LDA), Random Indexing [RJSK10], and most recently, word2vec [MCCD13] and GloVe [PSM14] are its successors. Nevertheless, the "basic" models, based on the Probabilistic Retrieval (PR) Framework [RZ+09], and Language Modeling (LM) [PC98] have maintained a respectable command of the research

and practice of IR. Despite the differences between the models, they are all fundamentally based on terms for establishing topical relevance relationships between documents and queries. A fundamental assumption in these models is the independence between terms in documents. One way to challenge this assumption is to go beyond the surface of terms and consider their semantics and relations in the document and eventually in IR models. Studying the approaches to combine the recent semantic models with IR models is the first research question of this thesis; Q1: *How can term associations be properly integrated in the PR Framework and LM while preserving their robustness and effectiveness?*

Integrating word embedding with IR models requires a deep understanding of its building block: term relatedness. While word embedding models promise a quantification of the similarity between terms, it is not clear to what extent this similarity value can be of practical use for distinguishing effective terms. Addressing this question has the potential to improve any other studies that use the related terms of word embedding models in text analysis tasks. Therefore, we put forward the second research question Q2: *Which range of similarity values is indicative of the actual term relatedness in document retrieval?*

While the effective selection of terms based on their similarities is a crucial step, calculating similarity still primarily depends on the underlying idea of creating semantic models: terms that share more common window-contexts (some terms around a term) tend to appear closer in the representation space and therefore are more probable to be considered as related. This is however not a sufficient assumption, especially for the needs of information retrieval, as for instance antonyms or co-hyponyms share common window-context—and therefore are considered as related—but can potentially bias the query to irrelevant topics. This issue raises the need to adapt the word embedding's relatedness in order to more effectively fit to IR tasks. To address it, we explore Q3: *Which other statistical subtleties can enrich window-context word embedding similarities for better IR performance, and how to exploit such information to effectively select relevant terms?*

So far the focus has been on the effectiveness of models. Beside performance, an important aspect of statistical models is whether they can be intuitively understood and interpreted. Interpretability particularly becomes increasingly important in data analysis and machine learning as the algorithms become more complicated and at the same time they gain greater roles in our life and society. In fact, it is crucial to know why a model suggests specific results and what the inference process is. In the context of word embedding models, despite their wide range of applications, the semantic vectors and the meanings of their corresponding dimensions remain difficult to interpret and hard to analyze. A natural way to make a semantic representation model understandable is by explicitly specifying the semantic concept that each dimension of its vectors refers to. Having such interpretable vectors would enable error resolution and better causal analysis, e.g. one can investigate the *reason* for similarity of two terms through tracing the matching dimensions with high values. In this thesis, we investigate this topic by exploring the question of Q4: *How to make word embedding vectors interpretable while preserving their effectiveness?*

The discussed topics address basic research questions, mainly in the framework of

document retrieval. The following topics study the application of the introduced methods in two other text analysis areas, namely cross-lingual word sense disambiguation, and sentiment analysis.

Word Sense Disambiguation (WSD) is the task of automatically selecting the most related sense for a word occurring in a context. Cross-Lingual Word Sense Disambiguation (CL-WSD) targets disambiguation of one word in a source language while translating to a target language. The approaches in (CL-)WSD typically require rich information resources such as structured knowledge resources or large amounts of annotated data i.e. parallel corpora (supervised methods). While these approaches usually achieve excellent results in practice, they have to face the knowledge acquisition bottleneck which is a particular problem in languages with scarce resources (low-density) like Persian. To address this problem, we approach CL-WSD using only information extracted from existing corpora as well as a simple dictionary (unsupervised method). In particular, the thesis explores Q5: *How can word embedding and semantic similarity of the terms in context be exploited for CL-WSD in the scenario of low-density languages, specially with the application of English to Persian?*

Finally, in the last research topic, we explore the use of word embedding-based IR models in sentiment analysis, and in particular, its application in financial volatility prediction. Volatility is an essential indicator of instability and risk in financial markets and has gained considerable attention during the last decades. Volatility prediction is typically approached using factual market data. However, the significant increase in the quantity and richness of textual data during the last years encourages the exploration of text analysis approaches. An interesting resource of textual information is the companies' annual disclosures, known as *10-K filing* reports. The reports contain comprehensive information about the companies' business as well as risk factors. Therefore, as the last research question in this thesis, we investigate Q6: *How can we exploit novel IR term weighting models based on word embedding for sentiment analysis of 10-K reports to effectively predict financial volatility?*

1.3 Contributions

Considering the above mentioned research questions, the main goal of this thesis is providing an in-depth understanding of semantic word representations in information retrieval by proposing novel IR models, representation space analysis, and word-context scope exploration. The thesis also contributes to the topic of word representation interpretability as well as the state-of-the-art of cross-lingual word sense disambiguation and document-based sentiment analysis. The proposed methods are applied on various domains such as news, health, finance, and social science. In the following, we review each of the contributions in detail.

As mentioned in Section 1.2, the first research question (Q1) concerns the integration of semantic models in the Probabilistic Relevance Framework by addressing term independence. To approach it, we consider a form of term-term relation, based on the

underlying concepts of each term. The concepts related to each term are extracted from an embedding model. Now in our novel retrieval models, instead of counting the occurrences of a term, the models count the occurrences of the term's concepts in the documents. We exploit this idea to revisit a wide spectrum of existing IR models, namely Pivoted Document Normalization, BM25, BM25 Verboseness Aware, Multi-Aspect TF, and Language Modeling. It turns out that this approach is in fact a generalization of the translation model [BL99] from Language Modeling to the PR Framework; and therefore we refer to the models as Generalized Translation models. In studying them, we observe a potential limitation of the translation models: they only affect the term frequency based components of all the models, ignoring changes in document and collection statistics. To correct this limitation, we propose extending the translation models with the statistics of term associations and provide extensive experimental results in Ad-hoc, news, and health domains to demonstrate the benefit of the newly proposed methods. Additionally, we compare the translation models with query expansion methods based on the same term association resources, as well as based on Pseudo-Relevance Feedback (PRF). We observe that Generalized Translation models always outperform the query expansion methods, but provide complementary information with PRF, such that by using PRF and our translation models together we observe results better than the current state of the art [RLHZ16].

The second contribution concerns the selection of related terms to a term in word embedding by analyzing similarity ranges (Q2). We hypothesize that the related words can be identified by a threshold on similarity values which separates the semantically related words from the non-related ones. To indicate such a threshold, we first observe and quantify the uncertainty of word embedding models with respect to the similarity values they generate. Based on this, we introduce a general threshold which effectively filters related terms. We particularly explore the effect of dimensionality on this general threshold by conducting the experiments in different vector dimensions. The effectiveness of the general threshold is evaluated on several Ad-hoc and news collections using various Generalized Translation models. The evaluation shows that using the proposed threshold leads to significantly better results than the baseline while being equal to, or statistically indistinguishable from, the best achieved results by parameter tuning [RLH17, RLH16].

As mentioned before, regardless of a high potential of word embedding as a resource for related terms, the incidence of several cases of topic shifting deteriorates the final performance of the applied retrieval models (Q3). To address this issue, we revisit the use of global context (i.e. the term co-occurrences in documents) to measure the term relatedness. We hypothesize that in order to avoid topic shifting among the terms with high word embedding similarity, they should often share similar global contexts as well. We therefore study the effectiveness of post filtering of related terms by various global context relatedness measures. Experimental results show significant improvements in two test collections, and support our initial hypothesis regarding the importance of considering global context in retrieval [RLHZ17].

The next topic contributes to the interpretability of word embedding models (Q4) by

transferring low-dimensional semantic vectors (dense vectors) of words to *explicit representations*. Explicit representations of words i.e. vectors with clearly-defined dimensions, which can be words, windows of words, or documents are easily interpretable, and recent methods show competitive performance to the dense vectors. In this contribution, we propose a method to transfer a state-of-the-art neural-based embedding model to its explicit representation model. The method provides interpretable explicit vectors while keeping the effectiveness of the original model, tested by evaluating the model on several word association collections. As a case study on the use of our explicit representation, we propose a novel method to quantify the degree of the existence of gender bias in the English language (used in Wikipedia) with regard to a set of occupations. By measuring the bias towards explicit Female and Male factors, the work demonstrates a general tendency of the majority of the occupations to male and a strong bias in a few specific occupations (e.g. nurse) to female [RMLH17].

As discussed before, the thesis also explores the application of the word embedding-based methods on CL-WSD and sentiment analysis (Q5). The proposed approach in CL-WSD exploits semantic similarity to find the best Persian translation of an ambiguous English term in an English sentence. In this approach, the method first uses a lexicon to translate the ambiguous English term to candidate Persian terms. It then calculates the semantic similarity values between each candidate and a Persian translation of the sentence, and finally selects the candidate term with the highest similarity value. The semantic similarity is calculated using a generated Persian word embedding model. We evaluate this approach on a recent evaluation benchmark and compare it to the state-of-the-art unsupervised system. The results show that the proposed method outperforms the state-of-the-art system in various evaluation metrics [RLHD17, RSL+16].

For sentiment analysis of financial reports mentioned in Q6, the thesis proposes the exploitation of the Generalized Translation models. To estimate the sentiment of a report, we use a lexicon of financial terms and calculate their weights in the report using the embedding-based translation models. By extensive evaluation of various sentiment analysis methods, we observe significant improvement with the proposed approach over state-of-the-art methods on volatility prediction accuracy. In addition, since factual market data have been widely used as the mainstream approach to forecast volatility, we study different fusion methods to combine text and market data resources. The final result achieves better performance than using each of the resources alone and shows the promising effectiveness of exploiting text resources for volatility prediction [RLB+17].

1.4 Structure of Thesis

The thesis is structured as follows: in Chapter 2, we explain in detail various algorithms to create word embedding models, followed by reviewing related studies.

The next six chapters respectively discuss the six research questions of the thesis (Q1-Q6), mentioned in Section 1.2.

The first four chapters (Chapter 3 to Chapter 6) discuss basic research on integration, adaptation, and interpretability of word embedding in IR. Chapter 3 studies the Generalized Translation models and reports their performances on document retrieval tasks. The novel translation models are then used in Chapter 4 to analyze the space of word embedding models in different dimensions and to propose a general threshold on similarity values for selecting the related terms. Chapter 5 continues this direction by examining the idea of combining window-context with global-context embeddings. In Chapter 6, we explore the interpretability of word embedding, followed by providing a case study on gender bias in Wikipedia.

The two chapters afterwards discuss the application of the introduced methods in two other text analysis tasks. Chapter 7 explains our unsupervised method for cross-lingual word sense disambiguation and evaluates it on the English-Persian test collection, described in Appendix A; Chapter 8 thoroughly studies sentiment analysis methods for financial volatility prediction using the Generalized Translation models.

Finally in Chapter 9, we conclude the thesis and discuss open research questions.

CHAPTER 2

Background and Related Work

This chapter provides an in-depth background on word representation models and their position in Information Retrieval. We start with studying various methods of creating word representations, followed by a comprehensive review on the state-of-the-art of applying word embedding in IR.

2.1 Word Representation Models

Word representation models and their applications has been a focus of NLP and IR communities for decades. In this section, we explain well-known and widely used methods for creating semantic word representations. As discussed before, the dimensions of the word vectors stand for some forms of semantic concepts in language which can be explicit (i.e. documents, or terms) or implicit (i.e. determined from the data, but not matching any existing terms or phrases).

In the following, we start with discussing an explicit representation model, followed by explaining two matrix factorization approaches for creating low-dimensional vectors. Two alternatives to matrix factorization are discussed afterwards: first an iterative method using random vectors, and then a prediction-based model based on neural networks. Finally, we discuss unobvious relations between the word representation models as well as their performance differences in practice.

Before explaining the word representation models, let us define some basic notations and operations: Any word representation model results to a set of vector representations of terms, denoted as V, in the size of $|W| \times d$ where W is the collection of terms in a language, and d is the dimension of the vectors. Using such vectors, one can calculate the semantic relation between two terms based on some measures of vector similarity/distance. We use the cosine function throughout the thesis:

$$sim(w, w') = cosine(w, w') = \frac{V_w \cdot V_{w'}}{|V_w|\,|V_{w'}|} \tag{2.1}$$

where V_w and $V_{w'}$ are the vectors of the terms w and w', respectively. While the use of cosine may be arguable, it is the current practice and an investigation in this sense is outside the scope of this thesis.

2.1.1 Explicit Word Representations: Point Mutual Information

Let us start with explicit word representation models, i.e. the models where each dimension of every term vector refers to a specific language entity e.g. a term in the short window-contexts around the original term, or a document containing the term. The explicit word representations could become very high-dimensional (upto the number of terms/documents in the collection), and also highly sparse (as an arbitrary term generally does not co-occur with many other terms or appear in many documents). Despite these facts, as mentioned in the introduction, explicit representation vectors have the benefit of being interpretable and are also the starting point for creating some of the low-dimensional word representations (discussed later in the section). It is therefore important to thoroughly discuss and understand them.

For the sake of brevity, in the following we only discuss the explicit semantic vectors based on window context terms (i.e. any term in the given window context of a term at hand), while the approaches can be easily generalized to bigger contexts i.e. paragraphs or documents.

Following this setting, in our explicit vectors, the number of dimensions is equal to the number of words in the collection: $d = |W|$. We define X as the set of all co-occurrence pairs (w, c), captured from window-contexts of the corpus, where $w \in W$ and $c \in W$ denote an arbitrary term and context term.

An explicit representation model uses the statistics extracted from X to define the co-occurrence relation between two terms, known as *first-order* or *syntagmatic* relation [SP93]. A well-known approach for the first-order relation is based on Point Mutual Information (PMI) [CH90, DPL94, NN94]. PMI measures how distinguishable is the probability of co-occurrence of two terms from their independent occurrence probabilities, defined as follows:

$$PMI(w, c) = \log \frac{p(w, c)}{p(w)p(c)} \tag{2.2}$$

where $p(w, c)$ is the probability of (w, c) in the co-occurrence collection: $\#(w, c)/|X|$ and $p(w)$ is the probability of the appearance of w with any other term: $\#(w, .)/|X|$ (same for $p(c)$).

A widely-used alternative is Positive PMI (PPMI) which replaces the negative values with zero:

$$PPMI(w, c) = max(PMI(w, c), 0) \tag{2.3}$$

Given either of the first-order relation methods, we define the explicit vector representation of term w as a vector with $|W|$ dimensions, where the value of each dimension is the PMI/PPMI between w and the corresponding term c of the dimension. As discussed at the beginning of the section, having vector representations of two terms (w and w'), we calculate the semantic similarity between the terms using Eq. 2.1. This semantic similarity is also known as *second-order* or *paradigmatic* relation between the terms.

2.1.2 Matrix Factorization: Latent Semantic Indexing, GloVe

As mentioned, the explicit word representations are generally in very high dimensions ($\sim 10K - 500K$ when terms as dimensions) and therefore inefficient for storing and computing. This can especially be a problem when the vectors are used as features for machine learning applications. Dimensionality reduction methods address this problem by providing low-dimensional or dense vectors ($\sim 10 - 1000$). Although the dimensions in dense word vectors are hardly interpretable, in practice they are more efficient than explicit word representations. In addition, due to the elimination of noise through dimensionality reduction processes, dense representations are in general expected to be more effective in downstream applications or term-similarity benchmarks.

In the text analysis community, one well-known approach for creating dense vectors is Latent Semantic Analysis/Indexing (LSA/LSI) [DDF$^+$90], initially applied on the term-document matrix and later on the term-term PMI matrix [Sch92]. To be consistent with the previous subsection, we only discuss the application of the LSI method on the term-term PPMI/PMI matrix, as applying it on the term-document matrix follows the same principle.

Calculating LSA is based on the *Singular Value Decomposition (SVD)*, a mathematical matrix factorization technique broadly used in signal processing and statistics. To create the Latent Semantic Index, SVD is applied on the PMI/PPMI matrix V (with size of $|W| \times |W|$), resulting to the following matrices:

$$V = U\Sigma C^T \tag{2.4}$$

where U is the term matrix with size of $|W| \times m$, m is the rank of V (number of linearly independent rows), Σ is an $m \times m$ diagonal matrix with singular values along the diagonal, expressing the importance of each dimension, and finally C^T is the context matrix in $m \times |W|$.

The singular values in Σ provide a quantification of the importance of each dimension, sorted from the most to least important one. To reduce dimensionality, we keep d (e.g. 300) top values on the diagonal of Σ and set the rest to zero. This method is referred to as truncated SVD where the parameter $d < m$ is the size of the embedding vectors. In practice, we simply truncate d first dimensions of the U matrix, resulting to a $|W| \times d$ words representation matrix.

Recently, Pennington et al. [PSM14] introduce the GloVe representation by following the idea of factorizing an explicit representation matrix to dense vectors. In GloVe, the explicit matrix is a term-term co-occurrence matrix Y of $|W| \times |W|$ size, factorized to two dense matrices: V, and $\widetilde{V}$, the set of term and context vectors, respectively. Both dense matrices are of $|W| \times d$ size and randomly initialized. GloVe aims to find some optimal values for the dense vectors by optimizing the following objective function:

$$J = \sum_{i=1}^{|W|} \sum_{j=1}^{|W|} f(Y_{ij})(V_{w_i}^T \widetilde{V}_{w_j} + b_i + \widetilde{b}_j - \log Y_{ij})^2 \tag{2.5}$$

where b_i and $\widetilde{b}_j$ are bias for the ith term, and jth context term respectively, and f is the weighting function, defined as follows, that assigns relatively lower weights to rare and frequent co-occurrences:

$$f(Y_{ij}) = \begin{cases} (\frac{Y_{ij}}{Y_{max}})^\alpha & \text{if } Y_{ij} < Y_{max} \\ 1 & \text{otherwise} \end{cases} \qquad Y_{max} = \max_{i,j} Y_{ij} \tag{2.6}$$

As suggested by the authors, the parameter α is set to 0.75 and the final representations are the average of the term vectors V and context vectors $\widetilde{V}$.

2.1.3 Iterative Vectors: Random Indexing

The introduced matrix factorization methods achieve the dense vectors from an explicit representation matrix, and in principle require large memory for storing the explicit matrix as well as high computational resources. As an alternative, Sahlgren [Sah05] introduces Random Indexing, a highly efficient method to generate dense word vectors. Random Indexing only stores two sets of dense vectors: *index vectors* and *context vectors*; and iteratively trains them in the following steps:

- In the first step, the index vectors are randomly generated and assigned to each context. These index vectors are sparse so that just a small number of their elements are randomly set to +1 or -1, and the rest are 0.
- Then, while reading the text, every time a given term occurs, the index vectors of the terms in its context are added to the context vector of the term. Terms are thus represented by context vectors that are effectively the sum of the terms' contexts.

The fact that Random Indexing only maintains two sets of low-dimensional matrices in the memory and does not require creating a large explicit matrix, provides considerable practical benefits. This is also a characteristic of the prediction-based methods, discussed below.

2.1.4 Prediction Instead of Counting: word2vec

Prediction-based representation models are rooted in the idea of language modeling: predicting the probability of occurrence of a term, given the observation of another term when they co-occur in a context window. Several studies have approached the estimation of these probabilities using neural networks techniques [BDVJ03, CW08]. More recently, Mikolov et al. [MSC+13] introduce word2vec, an efficient and effective neural network-based approach. The word2vec method suggests two prediction models: SkipGram, which predicts the context terms of a term from the occurrence of the term, and CBOW, which predicts the occurrence of a term, given its context terms. In the following, I only explain the SkipGram model as the CBOW is conceptually similar.

As in GloVe and Random Indexing, the SkipGram model starts with two randomly initialized sets of vectors: term (V) and context ($\widetilde{V}$) vectors, both of size $|W| \times d$. The objective of SkipGram is to find a set of V and $\widetilde{V}$—as the parameters of an optimization algorithm—by increasing the conditional probability of observing a context term c given another term w when they co-occur in a window, and decreasing it when they do not. In theory, this probability is defined as follows:

$$p(c|w) = \frac{\exp(V_w \widetilde{V}_c)}{\sum_{c' \in W} \exp(V_w \widetilde{V}_{c'})} \tag{2.7}$$

where as before, V_w and $\widetilde{V}_c$ are the term vector of the term w and the context vector of the term c, respectively.

Obviously, calculating the denominator of Eq. 2.7 is highly expensive and a bottleneck for scalability. One proposed approach for this problem is the *Noisy Contrastive Estimation (NCE)* [MT12] method. The NCE method, instead of computing the probability in Eq. 2.7, measures the probability which contrasts the *genuine* distribution of the term-context pairs (given from the corpus) from a *noisy* distribution. The noisy distribution $\mathcal{N}$ is defined based on the unigram distribution of the terms in the corpus. Formally, it defines a binary variable y, showing whether a given pair belongs to the genuine distribution: $p(y = 1|w, c)$. Further on, Mikolov et al. [MCCD13] proposed the *Negative Sampling* method by some simplifications in calculating $p(y = 1|w, c)$, resulting in the following formula:

$$p(y = 1|w, c) = \frac{\exp(V_w \widetilde{V}_c)}{\exp(V_w \widetilde{V}_c) + 1} = \sigma(V_w \widetilde{V}_c) \tag{2.8}$$

where σ is the sigmoid function ($\sigma(x) = 1/(1 + \exp(-x))$). Based on this probability, the cost function of the SkipGram method is defined as follows:

$$J = -\sum_{(w,c) \in X} \left[\log p(y = 1|w, c) + k \mathop{\mathbb{E}}_{\check{c}_i \sim \mathcal{N}} \log p(y = 0|w, \check{c}_i) \right] \tag{2.9}$$

where $\check{c}_i$ is each of the k sampled terms from the noisy distribution $\mathcal{N}$, X—as in Secion 2.1.1—is the set of all co-occurrences, and $\mathbb{E}$ denotes expectation value, calculated as the average for the k sampled terms.

In addition, two preprocessing steps dampen the dominating effect of very frequent terms: First is *subsampling* which randomly removes an occurrence of term w in the corpus when the term's corpus frequency $\#(w)$ is more than some threshold t, with a probability value of $1 - \sqrt{t/\#(w)}$. The second is *context distribution smoothing* (`cds`) which dampens the values of the probability distribution $\mathcal{N}$ by raising them to power $\alpha < 1$. Experimental results show an optimal value of $\alpha = 0.75$ for the SkipGram model which is the same as the optimal value of the α parameter in GloVe. Finally, suggested by the authors, only the V vectors are used for the word representations and the $\widetilde{V}$ set is discarded.

2.1.5 Relations and Comparison

Levy and Goldberg [LG14] show an interesting relation between PMI and SkipGram representations, i.e. when the dimension of the SkipGram vectors is set to very high (as in explicit representations), the achieved representation from the SkipGram objective function (Eq. 2.9) is equal to PMI shifted by $\log k$. They call this representation Shifted Positive PMI (SPPMI):

$$SPPMI(w, c) = max(PMI(w, c) - \log(k), 0) \tag{2.10}$$

They further integrate the ideas of subsampling and `cds` into SPPMI. Subsampling is applied during the creation of the X set by randomly removing very frequent words. The `cds` method adds a smoothing on the probability of the context term, as follows:

$$PMI_\alpha(w, c) = \log \frac{p(w, c)}{p(w)p_\alpha(c)} \quad p_\alpha(c) = \frac{\#(w, .)^\alpha}{\sum_{w' \in W} \#(w', .)^\alpha} \tag{2.11}$$

Interestingly, such explicit word representation demonstrates competitive performance to word2vec models when evaluated on various term-similarity benchmarks.

Comparing the subtleties as well as performance of the introduced semantics representation models has been the topic of several studies. Schnabel et al. [SLMJ15] report that despite sharing a common fundamental idea between various representation models, in practice, they show considerably different performance in downstream tasks such as sentiment classification, and noun phrase chunking. Baroni et al. [BDK14] evaluate the models for term-to-term similarity and report better performance of context-predicting methods (such as SkipGram) compared to the traditional context-counting methods. More recently, Levy et al. [LGD15] benchmark the models by taking into account their hyper-parameters as well as preprocessing steps. They show that there is no fundamental performance difference between the recent word embedding models and that the performance gain observed by one model or another is mainly due to the setting of the hyper-parameters of the models. They finally conclude: *"SkipGram is a robust baseline. While it might not be the best method for every task, it does not significantly underperform in any scenario."* This conclusion motivates us to focus on word2vec SkipGram as the studied representation model in the thesis.

2.2 Word Embedding in Information Retrieval

As mentioned in the introduction, a contribution of this thesis is the integration of word embedding in retrieval models based on the idea of translation models. Translation models were introduced by Berger and Lafferty [BL99] almost two decades ago as an extension to language modeling, specifically the Query Likelihood model [PC98]. In the Query Likelihood model, the score of a document d with respect to a query q is considered to be the probability of generating the query with a model M_d estimated based on the document:

$$score(q, d) = P(q|M_d) \tag{2.12}$$

The method to estimate $P(q|M_d)$ is therefore the essence here. This implies two issues: defining what kind of model M_d should be, and estimating the probability of q given the chosen model type. Typically, the model is a multinomial distribution and the probability is computed with a maximum likelihood estimator, together with some form of smoothing. Translation models as introduced by Berger and Lafferty essentially extend the $P(q|M_d)$ probability by including a translation probability P_T between all the terms t_d of the document d and each term t_q of the query:

$$P(q|M_d) = \prod_{t_q \in q} \left(\sum_{t_d \in d} P_T(t_q|t_d) P(t_d|M_d) \right) \tag{2.13}$$

This adds a third issue to the two above, the translation probability. Berger and Lafferty had used for computing P_T the Expectation Maximization approach inspired by machine translation approaches. Karimzadehgan and Zhai [KZ10] explore translation models using mutual information. Zuccon et al. [ZKBA15] use word2vec on translation language models, showing potential improvement in applying word embedding. Fang and Zhai [FZ06] explore the implementation of semantic matching for the axiomatic model, followed by Kraimzadehgan and Zhai [KZ12a] to extend it to translation models.

Other recent studies have combined language modeling and semantic word vectors: Ganguly et al. [GRMJ15] expand the classic language models through word embedding-based noisy channels which aim to discover the hidden dependencies between terms. Vulić and Moens [VM15] essentially provide a linear combination between language modeling and word embedding-based scores, calculated by generating an aggregated vector for the query. Tu et al. [TLLH14] directly apply a log-bilinear approach to learn semantic similarity into language modeling, expanding on previous work done by Wei and Croft [WC06] who used LDA and language modeling.

In general, while query likelihood models have demonstrated excellent performance in standardized benchmarking, a recurrent critique has been that they do not model the concept of relevance. Lafferty and Zhai [LZ03] introduced a formal way to relate language modeling to relevance, but the relation has been disputed by Robertson [Rob05] and others. Research in the context of the Probabilistic Relevance Framework has continued

in parallel to that on language modeling, with recently introduced models like the Multi Aspect TF (MATF) [Pai13], BM25 Verboseness Aware (BM25VA) [LLHA15].

There have been repeated efforts to expand methods of the probabilistic relevance framework with information about term-term relatedness. For instance, Zheng and Callan [ZC15] address query term weighting by exploiting word embedding as a feature vector to train a model for the optimal term weights. However, keeping the changes limited to the set of terms in the original query significantly limits the impact of their method. Zhao et al. [ZHY14] define a set of methods for distance-based cross term dependence and use them to modify the IR components, i.e. document term frequency, and document frequency, for boosting retrieval. The focus of their study was terms appearing in proximity of each other in terms of their locations in the documents, not in terms of their semantic representations. More recently, Lioma et al. [LSLH15] further explore the issue of identification of non-compositional phrases (i.e. a phrase with n terms, the meaning of which can not be explained by the composition of the meaning of the n terms), when they are composed of frequently co-occurring terms. These studies, addressing fundamentally the disadvantages of the unigram bag-of-words models are complementary to this work.

Expanding existing retrieval models with term-term similarity using translation models has an intuitive connection with direct query expansion methods, where terms are actually added to the query and/or weights are being recalculated. Xu and Croft [XC96], in one of the earlier papers in this area divide query expansion methods into *global techniques* and *local feedback*. That is, we can either use general knowledge about the terms, extracted from external resources such as logs [CWNM02, GN12], manual or automatic knowledge-bases [XJW09, KZ12b, XC15], word embedding models [ZC16, DMC16], or we can use some form of Pseudo-Relevance Feedback (PRF) [Roc71, LC01, ACR04].

For the global techniques, more than for the local feedback methods, attention has to be paid to the proper weighting of the new terms, as they come from outside the model used to rank documents. Cui et al. [CWNM02] and later Gao and Nie [GN12] use a logarithm to weight a term with respect to the query and the term-term similarities. Another way to define the weights on some set of candidate terms to be added to the query is by normalizing over all the added terms. This is done for instance by Xiong and Callan [XC15] when considering Freebase as a source of external knowledge and Zamani and Croft [ZC16] when using a word embedding model.

In terms of Pseudo Relevance Feedback (PRF), the probabilistic relevance framework has a built-in concept of relevance and therefore can naturally incorporate information provided through feedback [RZ+09]. For language modeling, Lavrenko and Croft [LC01] introduce the Relevance Model (RM), which selects expansion terms from top ranked documents and weights them based on the score of document ranking. The divergence from randomness (DFR) framework [AVR02] also allows a relatively straight-forward inclusion of feedback information: Amati et al. [ACR04] study the robustness of QE by two factors: divergence of the distribution of the query term in the retrieved documents from a random distribution and the frequency of the term in the whole document.

In a parallel line of work, word vectors are used for semantic similarity of two texts (i.e. paragraphs, documents, etc.). For instance, Kusner et al. [KSKW15] introduce Word Mover's Distance that is a direct extension of term-term similarity to text-text semantic similarity, applied on the vector representations of terms in documents. Their experiments show the effectiveness of the method on document classification tasks. However—to our knowledge—there is little evidence of robustness and effectiveness of such methods for retrieval tasks. In the next section, we introduce two text semantic similarity methods and briefly discuss their limitations for document retrieval.

Word embedding models have been also used as input for training neural retrieval models [SM15, GFAC16, XDC+17, MDC17]. These methods can be seen as the successors of the Learning to Rank models (a complete review at [L+09]), as they aim to model the concept of relevance in a supervised manner and usually require considerable amount of annotated data. In a recent study, Dehghani et al. [DZS+17] address the issue of need for sufficient training data by proposing the exploitation of top retrieved documents from established IR models.

Regarding the topic of exploration of word embedding space, Karlgren et al. [KBE+14] investigate the semantic topology of the Random Indexing vector space. Based on their previous observations that the dimensionality of the semantic space appears different for different terms [KHS08], Karlgren at al. now identify the different dimensionalities at different angles (i.e. distances) for a set of specific terms. They claim that in the embedding space "'close' *is interesting and* 'distant' *is not"* [KHS08]. In this thesis, we further explore this claim, focusing on IR-related criteria in vector space.

More recently, Gyllensten and Sahlgren [CGS15] follow a graph mining approach to represent the term relatedness by a tree structure and suggest traversing the tree as a potential approach for word sense induction tasks. They also point out that applying a nearest neighbor approach, where for every word we use the top k most similar words, is not theoretically justifiable.

In general, different characteristics of term similarities have been explored in several studies: the concept of relatedness [KB15, KHC15], the similarity measures [KZB+12], or intrinsic/extrinsic evaluation of the models [SLMJ15, TFL+15, BDK14, DVZK+14]. However, there is a lack of understanding on the internal structure of word embedding, specifically how the similarity distribution of representation vectors reflects the relatedness of terms. This is a contribution of this thesis.

While the mentioned studies generally use the existing word embedding models, some recent work focuses on training IR-specific representations. Diaz et al. [DMC16] suggest training separate word embedding models on a large set of top retrieved documents per query, while Zamani and Croft [ZC17] train query-level vectors using a smaller set of the top retrieved documents. This thesis follows these studies by exploring the effect of incorporating global context in word embedding similarity.

Exploiting global context for IR tasks has studied in several works: Tao and Zhai [TZ07] and later Lv and Zhai [LZ09] define measures of term proximity based on local and global

contexts, Bai et al. [BNCB07] combine global information with user profiles to specify the scope of queries, and Peterson et al. [PLSL15] propose novel local- and global-level coherence measures based on discourse entities and show the effectiveness of the coherence measures for document retrieval. Our work complements these studies by exploring the effects of combining word similarity achieved from global context with the window-context based similarity.

2.3 Summary

In this chapter, we provide the background on semantic representations and discuss related topics to the thesis. We first explain various word representation models, from explicit to neural network-based representation models. We discuss their characteristics as well as relations. Among them, based on a recent benchmark, we select word2vec SkipGram as the main word representation model, used in the rest of the thesis. We then review the related studies with special focus on the work about the exploitation of word embedding in information retrieval.

CHAPTER 3

Extended and Generalized Translation Models

In Information Retrieval, terms are still the fundamental building blocks for establishing topical relevance relationships between documents and queries. This is not a limitation of the research, nor of the machines, but rather a fact of human communication. We count terms because we cannot otherwise quantify meaning.

Despite the longstanding research on semantic models [DDF$^+$90, Hof99, BNJ03, RJSK10], the "basic" models based on the Probabilistic Retrieval (PR) Framework [RZ$^+$09], and language modeling [PC98] have maintained a respectable position in IR research. In spite of their differences in estimating probabilities, they are all fundamentally based on term frequency (tf) as a representation of the importance of a term within a document, and document frequency (df) as a representation of the specificity of a term, potentially normalized, pivoted, or smoothed by collection statistics (e.g. average document length, average term frequency, collection frequency).

The extension of these models with some form of semantic models receives continuous attention in IR community. Li and Xu [LX14] published a survey on the topic, grouping the various approaches into 5 categories:

1. Matching by Query Reformulation
2. Matching with Translation Model
3. Matching with Term Dependency Model
4. Matching with Topic Model
5. Matching with Latent Space Model

Both Topic Modeling and Latent Space Models are still to be conclusively proven competitive in terms of both efficiency and effectiveness with probabilistic and language models.

Term Dependency Models address one of the fundamental assumptions in IR, i.e. the occurrences of the terms in a document (or query) are independent from each other. Recently, Huston and Croft [HC14] presented a systematic comparison of such models. This line of research is complementary to the current study, as we consider the semantic relation of terms as a building block of our models.

Of the five categories, we focus here on Translation Models and Query Reformulations. The two are in fact related, because one may argue that a translation model acts as if the query had been reformulated. Both have a considerable history behind them. Considering Pseudo-Relevance Feedback (PRF) as a form of query reformulation, we can trace this back to the the late 60s [Roc71], while translation models have appeared immediately after the introduction of language models in the late 90s [BL99].

While translation models have been further investigated in the context of language modeling (reviewed in Section 2.2), the idea has not been considered in the context of the Probabilistic Relevance Framework [RZ+09]. In the context of the current advances of statistical semantic methods, it is therefore interesting to revisit these models, and potentially extend them towards the probabilistic models.

To address this, we propose to expand the PR Framework-based IR models in a way that does not affect their core tenets, but still takes advantage of the newly available, high-quality results in term-term similarity.

As in the classical PR Framework-based models, we consider the terms as the representations of concepts. A query "information management" is the composition of the two concepts denoted by the two terms. When to compute a tf/df score we count occurrences, we implicitly assume that a document containing the term "information" will be to some extent (proportional to the tf) about the concept denoted by this term. Equally, if the term "information" appears in many documents, we implicitly assume that it is not a discriminative term (proportional to df). A document containing the term "knowledge" however, is also related to the concept "information", yet it does not contribute to the sense of a document not containing "information". If we think of "information" however not as a term, but as a concept, we are entitled to replace the term "knowledge" with "information" and assign it a lower weight. It is here that the term-term similarity comes into play: the similarity is used to compute such a weight. Essentially, we are not even expanding the meaning of term frequency, because there was always an implicit assumption that we are counting concepts (this is why we normally do stemming). We propose to simply give tf the possibility to have fractional values above zero (instead of only natural numbers), when terms are conceptually related but are not the same.

This change, while coming from a different perspective on the nature of text documents, can be viewed as a generalization of the translation model idea from LM to the PR Framework.

However, when observed from the PR Framework perspective, this change has some implications on the other statistics used in IR models: document length, document frequency, and collection frequency. For instance, if we change the *tf*, then the length of a document, which is the sum of the *tf* values of its terms, changes as well. We set out to investigate the effects of these changes as well.

In summary, the main contributions of the current chapter are:

1. a generalization of the idea of translation models into the PR framework models (we consider four models: Pivoted Document normalization [SBM96], BM25 [RZ$^+$09], BM25 Verboseness Aware [LLHA15], and Multi-Aspect TF [Pai13])

2. an extension of the translation models in PR Framework by considering the effects of changing *tf* on all other term, document, and collection statistics.

3. extensive experimental results comparing the traditional translation model, the newly proposed ones, as well as query expansion methods, including Pseudo-Relevance Feedback.

The proposed models go beyond the state of the art in experimental results, and maintain the simplicity and robustness of the existing models, despite the fact that, we do not perform any optimization on existing parameters (e.g. b, k_1 in BM25).

The remainder of this chapter is structured as follows: First, we introduce the generalized as well as extended translation IR models in Section 3.1. Next, we present our experimental setup in Section 3.2, followed by discussing the results in Section 3.3. Section 3.4 summarizes our observations and concludes the chapter.

3.1 Novel Translation Models

We now introduce our approach to integrate the ideas of the translation model in the Probability Relevance Framework. We call it *Generalized Translation Model*. We put the focus of this study on four models: two classical: Pivoted Length Normalization [SBM96] and BM25, and two state-of-the-art schemes: Multi Aspect Term Frequency [Pai13] and BM25 Verboseness Aware [LLHA15].

While translation models only focus on changing the *tf* components, when we consider the relation between *tf* and other document and collection statistics in the probabilistic relevance framework, a valid hypothesis to investigate is that simultaneously changing the other components (e.g. *df*, document length) would further improve the final models. Our assumption is that these new models benefit from semantic relations of the terms while the robustness of the original models has been preserved. We call this approach *Extended Translation Model* and integrate it in the probabilistic relevance as well as the language modeling framework.

In what follows, first we explain the approach to extend the basic components of the models (tf, df) and then use the extended components to introduce the translation models in the four probability relevance models as well as in language modeling. Finally, we briefly revisit query expansion, explaining the approach for combining it with any translation model.

3.1.1 Basic Components

The fundamental idea of the introduced translation methods is, for each term t of a query q, to replace any existing related terms t' in a document d with the term itself, but counting its occurrence as a real number between zero and one. Consequently, a set of changes will appear in the definitions of tf_d, df, and T_d (term frequency, document frequency, and the set of terms in a document).

In order to define the new components, we first denote the set *related terms* to a given term as $R(t)$. The similarity value of each term in this set is expected to be between 0 and 1. As mentioned in Chapter 2, we calculate the value by using the cosine function of the vector representations of the terms from a word embedding model. To create this set, we follow two approaches: 1. using the top-N most similar terms and 2. filtering the terms with similarity values higher than a threshold. The details of each will be discussed in the next sections.

Let us start with T_d: the set of terms associated with a document d changes with respect to a query q by replacing each related term with the term of the query to which it is related:

$$\widehat{T}_d = T_d \setminus \bigcup_{t \in q} \{t' \in R(t)\} \cup \{t \in q \colon R(t) \cap T_d \neq \emptyset\} \tag{3.1}$$

As a consequence of this redefinition of the documents, we must change the document frequency statistic accordingly:

$$\widehat{df}_t = \left|\{d \in D \colon t \in T_d \vee \exists t' \in R(t), t' \in T_d\}\right| \tag{3.2}$$

where D is set of the documents in the collection. As defined here, the extended document frequency $\widehat{df}_t$ considers the documents containing similar words in addition to the ones with the term itself. The hypothesis is that it prevents over-scoring of the documents that have terms with many similar terms in the query.

Finally, and most importantly, given the set of the related terms to the query, we define the extended term frequency as follows:

$$\widehat{tf}_d(t) = tf_d(t) + \sum_{t' \in R(t)} P_T(t|t') tf_d(t') \tag{3.3}$$

As in translation models (Eq. 2.13), P_T is interpreted as the probability of observing term t, having observed term t'. Similar to Zuccon et al. [ZKBA15], we estimate this probability by using the semantic similarity of the two terms (Eq. 2.1). The new $\widehat{tf}_d(t)$

extends the basic $tf_d(t)$ by similar terms and therefore rewards the documents with more related terms.

Given the above three fundamental building blocks, the other remaining components are defined as follows:

$\widehat{L}_d = \sum_{t \in \widehat{T}_d} \widehat{tf}_d(t)$	document length
$\widehat{avgdl} = \frac{1}{\lvert D \rvert} \sum_{d \in D} \widehat{L}_d$	average document length
$\widehat{tf}_c(t) = \sum_{d \in D} \widehat{tf}_d(t)$	term collection frequency
$\widehat{L}_c = \sum_{t \in T} \widehat{tf}_c(t)$	collection size
$\widehat{avgtf}_d = \frac{1}{\lvert \widehat{T}_d \rvert} \sum_{t \in \widehat{T}_d} \widehat{tf}_d(t)$	average term frequency
$\widehat{mavgtf} = \frac{1}{\lvert D \rvert} \sum_{d \in D} \widehat{avgtf}_d$	mean average term frequency

where their original forms are denoted as L_d, $avgdl$, $tf_c(t)$, L_c, $avgtf_d$, and $mavgtf$ respectively.

3.1.2 Generalized and Extended Translation Models

Based on the extended factors we just defined, we revisit the IR models and replace their components with the introduced extended ones. Since the logarithm function is regularly used as the dampening function, we use $\Lambda(x) = log(1 + x)$ to shorten notations.

Pivoted Length Normalization

Singhal et al. [SBM96] identify a bias in the cosine normalization as it favors long documents in retrieval. They then propose the pivoted length normalization (PL) schema by introducing a correction factor on the document length normalization. By replacing the elements of the original model, we define the Generalized Translation model (GT) and Extended Translation (ET) model as follows:

$$PL_{GT}(q,d) = \sum_{t \in \widehat{T}_d \cap T_q} \frac{\Lambda(\Lambda(\widehat{tf}_d(t)))}{1 - s + s\frac{L_d}{avgdl}} tf_q(t) log \frac{|D| + 1}{df_t} \tag{3.4}$$

$$PL_{ET}(q,d) = \sum_{t \in \widehat{T}_d \cap T_q} \frac{\Lambda(\Lambda(\widehat{tf}_d(t)))}{1 - s + s\frac{\widehat{L}_d}{\widehat{avgdl}}} tf_q(t) log \frac{|D| + 1}{\widehat{df}_t} \tag{3.5}$$

We should note that the original formulation uses $1 + log(1 + log(tf_d))$ in the numerator, while in our formula above we use $log(1 + log(1 + tf_d))$. For values of $tf_d > 1$ there is little difference between the two variations, and they have both been used in the literature. In our case, as it is theoretically possible that $tf_d < 1$, the formulation $1 + log(tf_d)$ may give negative values, hence we prefer the $log(1 + tf_d)$ variant.

BM25

BM25 is a widely popular and well-studied weighting model, rooted in the 2-Poisson probabilistic model of term frequencies in documents [RZ+09]. The Generalized Translation model ($BM25_{GT}$) replaces the $\widehat{tf}_d(t)$ and $\widehat{T}_d$ components in the classical version:

$$BM25_{GT}(q,d) = \sum_{t \in \widehat{T}_d \cap T_q} \frac{(k_1+1)\overline{tf_d^{GT}(t)}}{k_1+\overline{tf_d^{GT}(t)}} \frac{(k_3+1)tf_q(t)}{k_3+tf_q(t)} \log \frac{|D|+0.5}{df_t+0.5} \tag{3.6}$$

with

$$\overline{tf_d^{GT}(t)} = \frac{\widehat{tf}_d(t)}{B(d)}, \quad B(d) = (1-b) + b\frac{L_d}{avgdl} \tag{3.7}$$

The extended version of the BM25 translation model is shown in Eq. 3.8 and 3.9:

$$BM25_{ET}(q,d) = \sum_{t \in \widehat{T}_d \cap T_q} \frac{(k_1+1)\overline{tf_d^{ET}(t)}}{k_1+\overline{tf_d^{ET}(t)}} \frac{(k_3+1)tf_q(t)}{k_3+tf_q(t)} \log \frac{|D|+0.5}{\widehat{df}_t+0.5} \tag{3.8}$$

$$\overline{tf_d^{ET}(t)} = \frac{\widehat{tf}_d(t)}{\hat{B}(d)}, \quad \hat{B}(d) = (1-b) + b\frac{\widehat{L}_d}{\widehat{avgdl}} \tag{3.9}$$

Multi Aspect TF

Recently, Paik [Pai13] addresses the limitations of the pivoted length normalization by exploiting new statistical factors in the Multi Aspect TF (MATF) schema. The first component is Term Frequency Factor (TFF) which consists of two factors: Relative Intra-document *tf* (RI) measures the importance of a term regarding to the average *tf* of the document and Length Regularized *tf* (LR) that considers the length of the document in relation to the average document length in the collection. Paik [Pai13] then mentions the different tendency of the factors to long and short queries and combines them using the parameter ω which promises a reasonable balance between the factors based on the query length. Both factors are dampened first by the *log* and then by $f(x) = \frac{x}{1+x}$. We therefore revisit the TFF component for the Generalized Translation model as follows:

$$\widehat{RI}(t,d) = \frac{\Lambda(\widehat{tf}_d(t))}{\Lambda(\widehat{avgtf}_d)} \tag{3.10}$$

$$\widehat{LR}_{GT}(t,d) = \widehat{tf}_d(t)\Lambda\left(\frac{avgdl}{L_d}\right) \tag{3.11}$$

$$\widehat{TFF}_{GT}(t,d) = \omega\frac{\widehat{RI}(t,d)}{1+\widehat{RI}(t,d)} + (1-\omega)\frac{\widehat{LR}_{GT}(t,d)}{1+\widehat{LR}_{GT}(t,d)} \tag{3.12}$$

As suggested by the paper, the ω parameter can be estimated by the following function where $|q|$ is the length of the query:

$$\omega = \frac{2}{1 + \Lambda(|q|)} \tag{3.13}$$

Respectively, the TFF component for the Extended Translation model is defined as follows:

$$\widehat{LR}_{ET}(t,d) = \widehat{tf}_d(t)\Lambda\left(\frac{\widehat{avgdl}}{\widehat{L}_d}\right) \tag{3.14}$$

$$\widehat{TFF}_{ET}(t,d) = \omega\frac{\widehat{RI}(t,d)}{1+\widehat{RI}(t,d)} + (1-\omega)\frac{\widehat{LR}_{ET}(t,d)}{1+\widehat{LR}_{ET}(t,d)} \tag{3.15}$$

The second component is the Term Discrimination Factor (TDC) which uses inverse document frequency as well as average elite set term frequency (AEF) based on the total occurrence of a term in the entire collection, defined as follows:

$$AEF(t) = \frac{tf_c(t)}{df_t} \tag{3.16}$$

$$TDC(t) = log\frac{|D|+1}{df_t}\frac{AEF(t)}{1+AEF(t)} \tag{3.17}$$

We formulate the extension of the factor as follows:

$$\widehat{AEF}(t) = \frac{\widehat{tf}_c(t)}{\widehat{df}_t} \tag{3.18}$$

$$\widehat{TDC}(t) = log\frac{|D|+1}{\widehat{df}_t}\frac{\widehat{AEF}(t)}{1+\widehat{AEF}(t)} \tag{3.19}$$

Finally, the Generalized and Extended MATF Translation models are defined by integrating the corresponding components:

$$MATF_{GT}(q,d) = \sum_{t\in\widehat{T}_d\cap T_q} \widehat{TFF}_{GT}(t,d)\,TDC(t) \tag{3.20}$$

$$MATF_{ET}(q,d) = \sum_{t\in\widehat{T}_d\cap T_q} \widehat{TFF}_{ET}(t,d)\,\widehat{TDC}(t) \tag{3.21}$$

BM25 Verboseness Aware

Most recently, Lipani et al. [LLHA15] address the document length normalization factor of BM25 by proposing a novel parameter-free length normalization method that removes the need for the b parameter of BM25, called BM25 Verboseness Aware (BM25VA). The method leverages the mean of the average occurrences of a term in the documents to discover and supervise the effect of verboseness in the documents. The BM25VA defines the B factor of the original BM25 model as follows:

$$B_{VA}(d) = mavgtf^{-2}\frac{L_d}{T_d} + (1 - mavgtf^{-1})\frac{L_d}{avgdl} \tag{3.22}$$

We define the Generalized BM25VA Translation model by replacing the B_{VA} with the $B(d)$ component of the Generalized BM25 Translation model (Eq. 3.6 and Eq. 3.7), shown in the following formulas:

$$BM25VA_{GT}(q,d) = \sum_{t \in \widehat{T}_d \cap T_q} \frac{(k_1+1)\overline{tf_d^{GT}(t)}}{k_1 + \overline{tf_d^{GT}(t)}} \frac{(k_3+1)tf_q(t)}{k_3+tf_q(t)} \log \frac{|D|+0.5}{df_t+0.5} \tag{3.23}$$

$$\overline{tf_d^{GT}(t)} = \frac{\widehat{tf}_d(t)}{B_{VA}(d)} \tag{3.24}$$

The Extended Translation model also replaces the B component of the Extended BM25 Translation model (Eq. 3.8 and Eq. 3.9), with a modified version of the $B_{VA}(d)$:

$$BM25VA_{ET}(q,d) = \sum_{t \in \widehat{T}_d \cap T_q} \frac{(k_1+1)\overline{tf_d^{ET}(t)}}{k_1 + \overline{tf_d^{ET}(t)}} \frac{(k_3+1)tf_q(t)}{k_3+tf_q(t)} \log \frac{|D|+0.5}{\widehat{df}_t+0.5} \tag{3.25}$$

$$\overline{tf_d^{ET}(t)} = \frac{\widehat{tf}_d(t)}{\widehat{B}_{VA}(d)} \tag{3.26}$$

where

$$\widehat{B}_{VA}(d) = \widehat{mavgtf}^{-2}\frac{\widehat{L}_d}{\widehat{T}_d} + (1 - \widehat{mavgtf}^{-1})\frac{\widehat{L}_d}{\widehat{avgdl}} \tag{3.27}$$

Language Model

The translation model has been introduced in the framework of language modeling [BL99], so in this case we only point out that the Generalized Translation model is the original one, as introduced by Berger and Laferty (i.e. it is Generalized from language modeling to the Probabilistic Relevance Framework). For completeness, we also introduce the Extended Translation model for the LM framework.

In order to unify the notation, we can rewrite the translation LM in Eq. 2.13 as follows:

$$LM_{GT}(q,d) = \prod_{t_q \in q} P_T(t|d) \tag{3.28}$$

where $P_T(t|d) = \sum_{t_d \in d} P_T(t_q|t_d)P(t_d|d)$ is the translation probability of generating term t in document d. Similar to related studies [KZ10, ZKBA15] , we define $P(t_d|M_d)$ as the maximum likelihood estimation and inject $P_T(t|d)$ into a Dirichlet smoothing function obtaining:

$$P_T(t|d) = \frac{L_d}{L_d + \mu} \left[\sum_{t' \in T_d} \frac{P_T(t|t')tf_d(t')}{L_d} \right] + \frac{\mu}{L_d + \mu} p(w|C) \tag{3.29}$$

Now we can select the alternative terms t' based on the set of related terms $R(t)$ and rewrite the element in the square brackets above by explicitly exposing the term t where its translation probability to itself is one:

$$tf_d(t) + \sum_{t' \in R(t)} P_T(t|t')tf_d(t') \tag{3.30}$$

Eq. 3.30 is in fact identical to our definition of $\widehat{tf}_d(t)$ (Eq. 3.3) and therefore we can formulate the translation language model based on $\widehat{tf}_d(t)$ factor as follows:

$$LM_{GT}(q,d) = \prod_{t_q \in q} \left(\sum_{t_d \in d} \frac{L_d}{L_d + \mu} \widehat{tf}_d(t) + \frac{\mu}{L_d + \mu} \frac{tf_c(t)}{L_c} \right) \tag{3.31}$$

Finally we define the Extended Translation model by replacing the other components with their extended versions:

$$LM_{ET}(q,d) = \prod_{t_q \in q} \left(\sum_{t_d \in d} \frac{\widehat{L}_d}{\widehat{L}_d + \mu} \widehat{tf}_d(t) + \frac{\mu}{\widehat{L}_d + \mu} \frac{\widehat{tf}_c(t)}{\widehat{L}_c} \right) \tag{3.32}$$

3.1.3 Translation Models with Query Expansion

Translation models have an intuitive connection with direct query expansion methods. A natural question arising from generalizing translation models into the probabilistic relevance framework is how they compare with query expansion methods and whether they benefit from pseudo relevance feedback (PRF).

Considering a query expansion method ϕ and the new set of terms as $\phi(q)$, the general query expansion models is defined as:

$$S^*(q,d) = \sum_{t \in \phi(q)} w_t(q)S(t,d) \tag{3.33}$$

where each of the new terms has a coefficient of w_t, S^* is the final document score, and S is a scoring schema. If ϕ is based on a word embedding model, then S must be one of the basic methods (i.e. not using either the Generalized, nor Extended Translation models) because we would be using the same set of terms in both cases. If ϕ is based a PRF method, then S can be any of the methods previously described because the set of terms would be with very high probability different.

When using word embedding for query expansion, $\phi(q)$ is defined as the union of the set of related terms to each query term:

$$\phi(q) = \bigcup_{t_q \in Tq} \{t \in R(t_q)\} \tag{3.34}$$

To define the weight of each expanded term $w_t(q)$ with word embedding, we follow the weighting models of two recent studies. The first model, used in Gao and Nie [GN12], applies the logarithm weighting. We call it *LOG*, defined as follows:

$$w_t(q) = ln \left(\prod_{t_q \in Tq} P_T(t|t_q) + 1 \right) \tag{3.35}$$

The second expansion model [XC15] normalizes the P_T value (term-term similarity) over the sum of all expanded terms' similarities. We refer to this model as *NORM*, defined in the following formula:

$$w_t(q) = \frac{\sum_{t_q \in Tq} P_T(t|t_q)}{\sum_{t' \in \phi(q)} \sum_{t_q \in Tq} P_T(t'|t_q)} \tag{3.36}$$

Both the LOG and NORM expansion models calculate the final score using Eq. 3.33, while their weighting methods are only for the expanded terms and the weights of the original terms of the query are one ($t \in T_q : w = 1$).

3.2 Experiment Setup

In order to evaluate the performance of the introduced Generalized Translation (GT) and the Extended Translation (ET) models, we evaluate them based on each of the mentioned relevance models (Section 3.1.2) on six test collections. In addition, we combine and test both translation models with the PRF query expansion method as described in Section 3.1.3. We denote the Generalized Translation and Extended Translation models, combined with PRF as *PRF-GT* and *PRF-ET* respectively.

In the following, we introduce our experimental methodology, including test collections, baselines, parameter settings, and evaluation metrics.

Data Resources We conduct the experiments on six collections: combination of TREC 1 to 3 (TREC 123), TREC 6, TREC 7, and TREC 8 of the Ad-hoc track, TREC 2005 HARD track, and CLEF eHealth 2015 Task 2 User-Centred Health Information Retrieval [PZG+15]. For the TREC tasks we always used the title of the queries for retrieval. Table 3.1 summarizes the statistics of the test collections. For pre-processing, we apply the Porter stemmer and remove stop words using a small list of 127 common English terms.

Table 3.1: Test collections for evaluating the Generalized and Extended Translation models

Name	Collection	# Doc	Topics
TREC 123	Disc1&2	740088	51-200 Ad-hoc
TREC 6	Disc4&5	551873	301-350 Ad-hoc
TREC 7, 8	Disc4&5 without CR	523951	351-400, 401-450 Ad-hoc
HARD	AQUAINT	1033461	2005 Track (50 topics)
eHealth	as defined in [PZG+15]	1104337	CLEF-eHealth 2015 Task 2 (67 topics)

We train the word embedding model for the Ad-hoc and Hard tracks using the Wikipedia dump file for August 2015. For the eHealth task, similar to Koopman et al. [KZB+12] that train word embeddings based on the domain corpora, we use the corpus extracted from the task's collection. For both the word embeddings, we use the word2vec SkipGram method with vectors of 300 dimensions, sub-sampling parameter set to 10^{-5}, context windows of 5 terms, epochs of 25, and term count threshold 20. Our own experiments (not reported here) as well as those reported by Zuccon et al. [ZKBA15], indicate these parameters as reasonable as a general baseline.

Baselines In order to test the performance of the introduced translation models (GT and ET), for each IR schema we define three baselines: *STD* (the original version of the models), LOG (Eq. 3.35), and NORM (Eq. 3.36). In addition to LOG and NORM expansion methods, we experiment with the direct use of translation probability $P_T(t|t_q)$ as the weight for expansion. However due to the extremely weak performance observed, we remove it from the baselines.

In the experiments also using Pseudo Relevance Feedback query expansion (PRF-GT and PRF-ET), we test them against two baselines: original PRF, and original model (STD).

Finally, in both basic and with PRF modes, we test the performance of Extended Translation model (ET/PRF-ET) against the Generalized Translation model (GT/PRF-GT) respectively.

All the baselines as well as their corresponding symbols for the significance test are summarized in Table 3.2. Statistical significance tests are done using the two sided paired t-test and statistical significance is reported for $p < 0.05$.

Related Terms An essential part of all our extended models is the definition of "the set of related terms". In order to find this set for a given term ($R(t)$ in Section 3.1.1), we consider two approaches: 1. selecting the top-N similar terms in the collection, and 2. selecting the set of terms whose similarity values to the term t are above a threshold θ.

Normally, the first approach is the common method for defining the related terms, used in several studies [ZKBA15, GRMJ15]. However, as shown by Karlgren et al. [KHS08],

Table 3.2: Baselines and their symbols for the significance tests

Baseline	Tested from	Sig. Test
STD	All the models and baselines	†
LOG	GT, ET	ℓ
NORM	GT, ET	ν
PRF	PRF-GT, PRF-ET	ρ
GT / PRF-GT	ET / PRF-ET	§

the distribution of the distances of the neighboring terms is different for various terms, i.e. some words have more/less neighbors in a specific boundary. Inspired by this study, we observe the neighboring terms of the term 'excursion' in the word2vec model and spot the term 'tourist' is the 4th most similar (closest) neighbor. However, looking at the top neighbors of 'tourist', the term 'excursion' is the 17th one[1]. We assume that as 'tourist' is a more frequent term with more contexts in the language, its neighborhood is richer than 'excursion' and in other words has more related terms. This observation motivates us to investigate the effect of selecting the related terms based on a threshold θ in addition to the top-N approach.

Parameter Setting Since the basic parameters of each model are shared between the Generalized/Extended Translation models and original methods, the choice of parameters is not explored in this study and a standard set of parameters is used. For BM25 and BM25 Verboseness Aware (BM25VA), we set $b = 0.6$ (only for BM25), $k_1 = 1.2$, and $k_3 = 1000$, for Pivoted Length Normalization (PL), the parameter s is set to 0.05, and for Language Modeling (LM), we set μ to 1000. The Multi Aspect TF (MATF) does not require any parameter setting. For PRF we arbitrarily fixed the number of top-ranked documents to 3 and the number of expanded terms to 10.

In filtering related terms, for the top-N approach, following the related studies, we try N with 2, 5 and 10. For the threshold approach, we simply choose $\theta = 0.7$ for all the collections as a generally stable threshold parameter. The full discussion about the reason behind this choice as well as the stability in retrieval performance using this parameter is presented in the next chapter.

Evaluation Metrics The evaluation of retrieval effectiveness is done with respect to MAP and NDCG@20, as standard measures. However, our initial experiments showed that the extended methods retrieved a substantial proportion of unjudged documents. Looking at some of these unjudged retrieved results, we find different documents that seem relevant to the query. For example, as shown in Table 3.3, the document does not contain the term 'espionage' requested by the query, but there are many occurrences of the similar words like 'spy', 'intelligence service', or 'agent'. We assume that it is

[1]Noted that, the Cosine similarity is symmetric and in this case its values is equal to 0.54.

Table 3.3: Example of conceptually related document, found by our approach, while not judged in the TREC 6 Ad-hoc track

Query 311	Industrial Espionage
Document FBIS4-23903	... recruited last year by the **intelligence** service once more ... were indicted for high treason in the form of **spying**, including ... **agent** was in particular in charge of financing ...

due to the essential difference between the extended models and the standard term frequency-based methods which contributed to the creation of the relevance assessments used in the collections. Therefore, in order to provide a fairer evaluation framework, we consider MAP and NDCG over the condensed lists, which are proposed as better solutions to the incompleteness problem than BPREF in [Sak07][2].

3.3 Results and Discussion

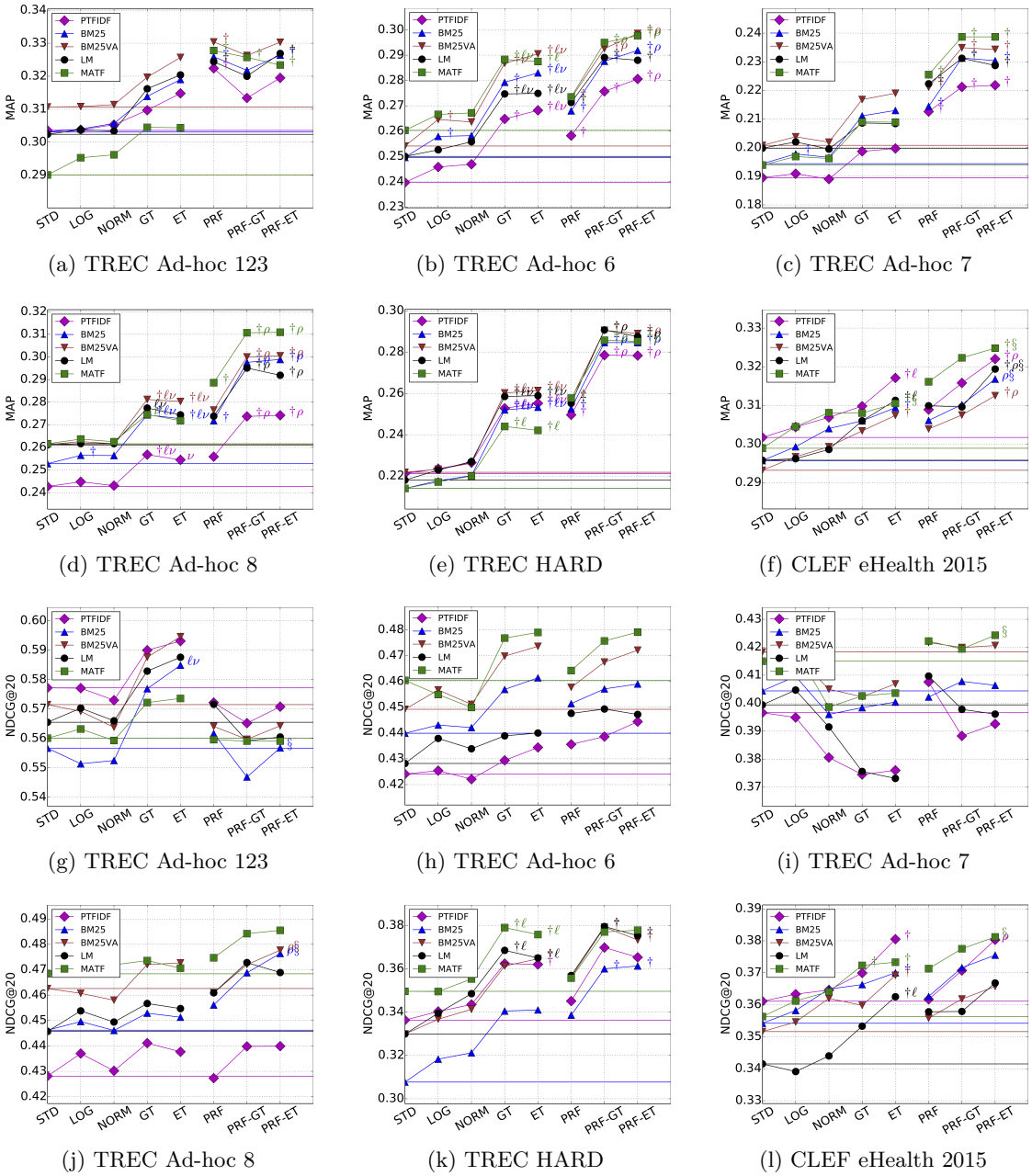
We evaluate the performance of the introduced Generalized as well as Extended Translation models on the mentioned IR models (Section 3.1.2) with $\theta = 0.7$, as discussed in the previous section.

The evaluation results of the MAP and NDCG@20 measures on the six test collections are shown in Figure 3.1. Each line in the plots shows the result of one IR model in two sections: from STD to ET the standalone version, and from PRF to PRF-ET when combined with the Pseudo Relevance Feedback query expansion. Significant differences of the results against the respective baselines are marked on the plots using the symbols, defined in Table 3.2. Table 3.4 shows the detailed results.

Starting with the results of the MAP measure, we observe that using the Generalized as well as Extended Translation models we gain significantly better performance in 4 of 6 collections, compared to the original models as well as compared with the LOG and NORM expansion methods. Only in the TREC 123 and TREC 7 collections, there is no statistically significant improvement, although there is no deterioration of results either. Looking at the expansion methods, the LOG and NORM models also improve the baseline only slightly.

The results of combining PRF query expansion with the Generalized and Extended Translation models shows significant improvement over the original as well as PRF models (except in TREC 123 and TREC 7), achieved by both translation models. This improvement over PRF is similar to the improvement achieved by the models without PRF over the original models, showing indeed that *global techniques* and *local feedback* can effectively complement each other.

[2]The condensed lists are used by adding the -J parameter to the trec_eval command parameters



(a) TREC Ad-hoc 123 (b) TREC Ad-hoc 6 (c) TREC Ad-hoc 7

(d) TREC Ad-hoc 8 (e) TREC HARD (f) CLEF eHealth 2015

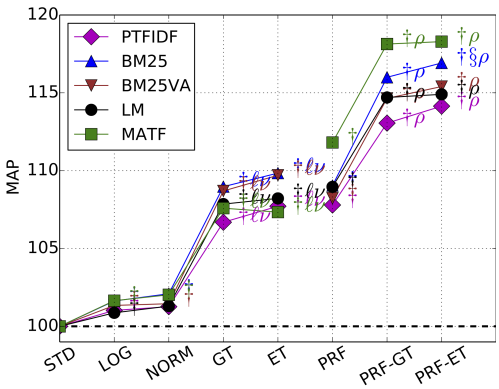
(g) TREC Ad-hoc 123 (h) TREC Ad-hoc 6 (i) TREC Ad-hoc 7

(j) TREC Ad-hoc 8 (k) TREC HARD (l) CLEF eHealth 2015

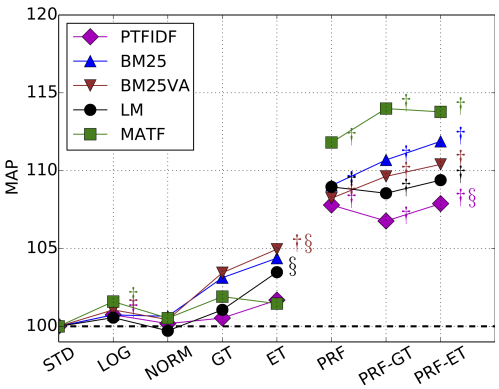
Figure 3.1: MAP and NDCG@20 evaluation of the TREC 123, TREC 6, TREC 7, TREC 8 Ad-hoc, TREC 2005 HARD, and CLEF eHealth 2015 task 2. The baselines and the signs for significance difference tests are shown in Table 3.2. The related terms are filtered when the similarities of the neighboring terms are higher than the threshold $\theta = 0.7$

Table 3.4: MAP and NDCG@20 evaluation of the TREC 123, TREC 6, TREC 7, TREC 8 Ad-hoc, TREC 2005 HARD, and CLEF eHealth 2015 task 2. In the models that need a set of related terms, the set is calculated based on the threshold approach with $\theta = 0.7$. The corresponding baselines for each model and their signs for the test of significance are shown in Table 3.2.

Collection	Method	PTFIDF		BM25		BM25VA		LM		MATF	
		MAP	NDCG	MAP	NDCG	MAP	NDCG	MAP	NDCG	MAP	NDCG
TREC 123	STD	0.304	0.577	0.303	0.557	0.311	0.572	0.302	0.566	0.290	0.560
	LOG	0.304	0.577	0.304	0.551	0.311	0.569	0.304	0.570	0.295	0.563
	NORM	0.305	0.573	0.306	0.552	0.311	0.564	0.303	0.566	0.296	0.559
	GT	0.310	0.590	0.314	0.577	0.320	0.588	0.316	0.583	0.304	0.572
	ET	0.315	0.593	0.319	0.585$\ell\nu$	0.326	**0.595**	0.320	0.588	0.304	0.574
	PRF	0.322†	0.572	0.326†	0.562	**0.330**†	0.564	0.324†	0.572	0.328†	0.560
	PRF-GT	0.313	0.565	0.322	0.547	0.326	0.560	0.320	0.559	0.326†	0.559
	PRF-ET	0.320	0.571	0.326†	0.557§	0.330	0.564	0.327†	0.560	0.323†	0.559
TREC 6	STD	0.240	0.424	0.250	0.440	0.254	0.449	0.250	0.428	0.260	0.460
	LOG	0.246	0.425	0.258†	0.443	0.265†	0.457	0.253	0.438	0.267	0.455
	NORM	0.247	0.422	0.258	0.442	0.264	0.451	0.256	0.434	0.267	0.450
	GT	0.265†	0.429	0.279†	0.457	0.287†ν	0.470	0.275†$\ell\nu$	0.439	0.288†$\ell\nu$	0.477
	ET	0.268†$\ell\nu$	0.434	0.283†$\ell\nu$	0.461	0.290†$\ell\nu$	0.474	0.275†$\ell\nu$	0.440	0.287†ℓ	0.479
	PRF	0.258†	0.436	0.268†	0.451	0.273†	0.458	0.271†	0.448	0.274	0.464
	PRF-GT	0.276†	0.439	0.288†	0.457	0.293†ρ	0.468	0.289†	0.449	0.295†ρ	0.476
	PRF-ET	0.281†ρ	0.444	0.292†ρ	0.459	**0.299**†ρ	0.472	0.288†	0.447	0.298†ρ	**0.479**
TREC 7	STD	0.190	0.397	0.194	0.404	0.201	0.418	0.200	0.399	0.194	0.415
	LOG	0.191	0.395	0.198†	0.410	0.204	0.417	0.202	0.405	0.197	0.419
	NORM	0.189	0.381	0.197	0.396	0.202	0.405	0.200	0.392	0.196	0.399
	GT	0.199	0.374	0.211	0.398	0.217	0.402	0.209	0.376	0.209	0.403
	ET	0.200	0.376	0.213	0.400	0.219	0.407	0.208	0.373	0.209	0.404
	PRF	0.213†	0.408	0.214†	0.402	0.221†	0.422	0.222†	0.410	0.226†	0.422
	PRF-GT	0.221†	0.388	0.231†	0.408	0.235†	0.420	0.231†	0.398	**0.239**†	0.419
	PRF-ET	0.222†	0.393	0.230†	0.406	0.234†	0.421	0.229†	0.396	**0.239**†	**0.424**§
TREC 8	STD	0.243	0.428	0.253	0.446	0.261	0.463	0.261	0.446	0.262	0.468
	LOG	0.245	0.437	0.257†	0.450	0.263	0.461	0.262	0.454	0.264	0.476
	NORM	0.243	0.430	0.256	0.446	0.262	0.458	0.262	0.449	0.263	0.472
	GT	0.257†$\ell\nu$	0.441	0.274†$\ell\nu$	0.453	0.281†$\ell\nu$	0.472	0.277$\ell\nu$	0.457	0.275ν	0.474
	ET	0.255ν	0.438	0.273†$\ell\nu$	0.451	0.280†$\ell\nu$	0.473	0.274	0.455	0.272	0.471
	PRF	0.256	0.427	0.272†	0.456	0.277	0.461	0.274	0.461	0.289†	0.475
	PRF-GT	0.274†ρ	0.440	0.298†ρ	0.469	0.300†ρ	0.472	0.295†ρ	0.473	0.311†ρ	0.484
	PRF-ET	0.274†ρ	0.440	0.299†ρ	0.476ρ§	0.300†ρ	0.478ρ§	0.292†ρ	0.469	**0.311**†ρ	**0.485**
HARD	STD	0.221	0.336	0.214	0.308	0.222	0.330	0.218	0.330	0.214	0.350
	LOG	0.224	0.340	0.218	0.318	0.223	0.337	0.223	0.340	0.217	0.349
	NORM	0.227	0.344	0.220	0.321	0.226	0.341	0.227	0.348	0.220	0.355
	GT	0.253†$\ell\nu$	0.362	0.252†$\ell\nu$	0.341	0.260†$\ell\nu$	0.361	0.259†$\ell\nu$	0.368†ℓ	0.244†ℓ	0.379†ℓ
	ET	0.255†$\ell\nu$	0.362†	0.253†$\ell\nu$	0.341	0.261†$\ell\nu$	0.365†	0.259†$\ell\nu$	0.365†ℓ	0.242†ℓ	0.376†ℓ
	PRF	0.250†	0.345	0.253†	0.339	0.257†	0.356	0.255†	0.357	0.258†	0.356
	PRF-GT	0.279†ρ	0.370	0.285†ρ	0.360†	0.290†ρ	0.379†	**0.291**†ρ	**0.380**†	0.286†ρ	0.377
	PRF-ET	0.278†ρ	0.365	0.285†ρ	0.361†	0.289†ρ	0.373†	0.287†ρ	0.375†	0.285†ρ	0.378
eHealth	STD	0.302	0.361	0.296	0.354	0.293	0.352	0.296	0.342	0.299	0.356
	LOG	0.304	0.363	0.299	0.358	0.297	0.355	0.296	0.339	0.304	0.361
	NORM	0.307	0.365	0.304	0.365	0.299	0.362	0.299	0.344	0.308	0.364
	GT	0.310	0.370	0.306	0.366	0.303	0.360	0.306	0.353	0.308	0.372†
	ET	0.317†ℓ	0.381†	0.309†	0.370†	0.307†	0.369†	0.311†ℓ	0.362†ℓ	0.310†§	0.373†
	PRF	0.309	0.362	0.306	0.363	0.304	0.356	0.310	0.358	0.316	0.371
	PRF-GT	0.316	0.371	0.310	0.372	0.307	0.362	0.310	0.358	0.322	0.378
	PRF-ET	0.322†ρ	0.380ρ	0.317ρ§	0.376	0.312†ρ	0.366	0.319†ρ§	0.367	**0.325**†§	**0.381**§



(a) Related terms with best performing threshold ($\theta = 0.7$)



(b) Related terms with best performing top-N ($N = 2$)

Figure 3.2: The gain of the models with the MAP measure regarding to their original versions, aggregated over all the collections.

Comparing the Extended Translation model with the Generalized one, in general ET/PRF-ET brings only a slight improvement to GT/PRF-GT. In some cases, notably the eHealth collection, the PRF-ET model provides a significant improvement over all the other models including PRF-GT.

The trends in the results of the NDCG@20 measure are generally similar to the ones of MAP, except in some rare cases such as the LM and PTFIDF methods in the TREC 7 collection.

In order to have an overview on all the models, we calculate the gain of each model over its original form and average the gains on the six collections. As the results for MAP are depicted in Figure 3.2a, GT and ET show significance improvement over the baselines. Also, while PRF has improved the baselines, its performance has then significantly been boosted by the Generalized and Extended Translation models. In addition, ET/PRF-ET show overall slight improvement to GT/PRF-GT. In some cases, e.g. for the BM25 and BM25VA models, this is significant.

In order to compare with previously reported results, Table 3.5 shows the best achieved results in each collection with the typical evaluation (i.e. not considering only the condensed lists, but rather considering the retrieved unjudged documents as non-relevant). Identifying the state-of-the-art for each collection by reviewing the literature is difficult and potentially controversial. TREC 8 Ad-hoc is however one of the most widely reported benchmarks, and regardless of whether we consider the condensed lists or not, the generalized and extended translation models proposed here show considerable improvements with respect to reports of the most recent experiments in our field [GRMJ15, Pai13, ZKBA15, LLHA15].

Table 3.5: The best results per collection

Collection	Eval. Measure	Method	Scoring	Value
TREC 123	MAP	PRF	BM25V	0.306
	NDCG@20	ET	BM25V	0.571
TREC 6	MAP	PRF-ET	BM25V	0.270
	NDCG@20	PRF-ET	BM25V	0.455
TREC 7	MAP	PRF-ET	MATF	0.226
	NDCG@20	PRF-ET	MATF	0.424
TREC 8	MAP	PRF-ET	MATF	0.295
	NDCG@20	PRF-ET	MATF	0.481
HARD	MAP	PRF-GT	BM25V	0.241
	NDCG@20	PRF-GT	BM25V	0.375

Threshold or Top-N As mentioned before, we considered two approaches for selecting the related terms: threshold-based and top-N. Figure 3.2b shows the aggregated gain of the best performing top-N approach ($N = 2$) over all the collections. Comparing it with Figure 3.2a, we see that while the selection of related terms from the top N terms generally improves the baselines, the performance of GT and ET and respectively PRF-GT and PRF-ET models using the threshold method considerably outperforms the top-N approach.

By having a closer look at the number of selected terms per term in the threshold approach with $\theta = 0.7$, we see a wide range of numbers, from 0 (no expansion) in several cases to a maximum of 63 terms. The average number of terms is 1.4, but the standard deviation is 3.7.

On the other hand, the LOG and NORM models are only marginally affected by changing the approaches and keep the results close to the baseline. This is due to their conservative approaches for weighting the expanded terms—aggregating over all weights in NORM and dampening in LOG.

Limitations As with any method relying on a numerical value to represent the similarity of two terms, our extended components are limited by the definition of similarity. Analysing the cases where the extended model results were lower than the optimal showed that sometimes the extended terms introduce bias in search as they represent related terms but not similar ones. For example, the word embedding models indicate 'Alzheimer' as highly related to 'Multiple sclerosis (MS)' (as they usually appear in very similar contexts), although they are not similar in the sense that a query on one of them is hardly presumed to be satisfied by a document on the other. However, this is a general issue in query expansion, when the expanded words introduce bias to the original query. We address some of these limitations in Chapter 5 of the thesis.

Efficiency Before concluding, it is worth noting that the Generalized as well as the Extended Translation models do not impose significant query-time overheads on the existing IR engines. Given the threshold, the set of related terms can be precomputed. The overhead of changing the statistics of the collection for the Extended model is computationally similar to the query time which makes it similar to the overhead of using PRF. Further optimization in this area is certainly possible. An implementation of the novel translation models for Solr and Lucene is available on Github[3].

3.4 Summary

In this chapter, we propose a generalization and an extension of translation models in the probabilistic relevance framework models in order to take advantage of word representation resources.

Concretely, we introduce changes in the calculation of core elements of probabilistic relevance framework models (term frequency, document frequency), following the implicit assumption that query terms denote concepts and that counting the presence of these terms in the documents and the collection is a surrogate for counting the presence of the concepts. By simply replacing the occurrence of similar terms with that of the query terms we maintain the simplicity and robustness of the existing models, while improving retrieval performance. We compare this approach with query expansion and also combine it with PRF based methods, observing the complementary effect of these two approaches, resulting in boosted performance.

This improvement in retrieval effectiveness is demonstrated on six test collections and five IR models, by achieving state-of-the-art results.

In the process, we also observe the effectiveness of selecting the "related terms" based on similarity boundary around the neighboring space of a term. This approach shows competitive performance compared with selecting the top-N most similar terms.

[3] `https://github.com/sebastian-hofstaetter/ir-generalized-translation-models`

CHAPTER 4

Similarity Threshold for Terms Relatedness

As discussed in the introduction, word embedding methods provide vectors that are proxies to the meaning of terms, and their semantic similarities. Fundamentally, word embedding models exploit the contextual information of the target terms to approximate their meaning, and hence their relations to other terms.

Given the vectors representing terms, these models provide an approximation of the similarity of any two terms, although this similarity relation could be perceived as completely arbitrary in the language. In this chapter, we address this issue by exploring how to identify whether the similarity score obtained from word embedding is actually indicative of term relatedness.

We hypothesize that the "similar" terms can be identified by a threshold on similarity values which separates the semantically related terms from the non-related ones. This threshold is general for the word embedding model and defined on all the terms. Using such a threshold, each term selects a number of similar terms, positioned in its neighborhood space.

Such a threshold has the potential to improve all studies that use similar/related terms in different tasks i.e. query expansion [GDR$^+$15], query auto-completion [Mit15], document retrieval [ZKBA15], learning to rank [SM15], language modelling in IR [GRMJ15], or Cross-Lingual IR [VM15]. It should be noted though, that the meaning of "similar" also depends on the similarity function. As mentioned before, in this thesis we consider the cosine function as it is by far the most widely used term similarity function and leave the exploration of other functions for further studies. In fact, regardless of the similarity function, a threshold that separates the semantically related terms from the rest will always be an essential element to identify.

We explore the estimation of this potential threshold by first quantifying the uncertainty in the similarity values of embedding models. This uncertainty is an intrinsic characteristic of the recent models, because they all start with some random initialization and eventually converge to a (local) solution. Therefore, even by training with the same parameters and on the same data, the created word embedding models result in slightly different term distributions and hence slightly different relatedness values. In the next step, using this observation, we provide a novel representation on the expected number of neighbors of an arbitrary term as a continuous function over similarity values, which is later used to estimate the general threshold.

In order to evaluate the effectiveness of the proposed threshold, we use the novel translation models, introduced in Chapter 3, and test them on four test collections. In the experiments, we apply the threshold to identify the set of terms to extend the query terms using both the Generalized Translation Model and the Extended Translation Model. In fact, we follow the study in the previous chapter by exploring an effective threshold value via studying the uncertainty in the embedding space. The results of using the proposed threshold are compared with the optimal threshold, achieved—as before—by exhaustive search on the spectrum of threshold parameters. We show that in general using the proposed threshold performs either exactly the same as, or statistically indistinguishable from the optimal threshold.

In summary, the main contributions of this chapter are:

1. exploration of the uncertainty in word embedding models in different dimensions and similarity ranges.

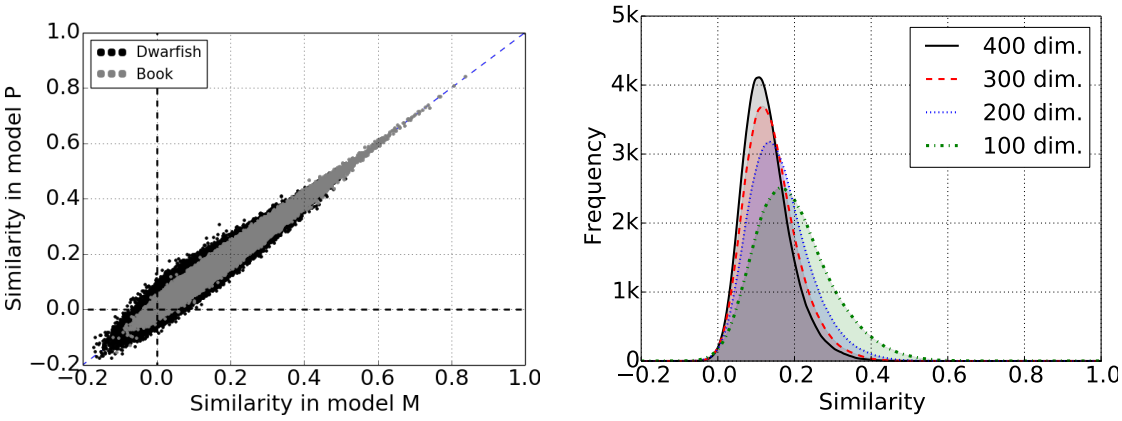
2. introducing a general threshold for separating similar terms in different embedding dimensions.

3. extensive experiments on four test collections comparing different threshold values on different retrieval models.

The remainder of the chapter is structured as follows: We introduce the proposed threshold in Section 4.1. We next present our experimental setup in Section 4.2, followed by discussing the results in Section 4.3. Section 4.4 summarizes our observations and concludes the study.

4.1 Global Term Similarity Threshold

We are looking for a threshold to separate the related terms from the rest. For this purpose, we start with an observation on the uncertainty of similarity in word embedding models, followed by defining a novel model of the expected number of neighbors for an arbitrary term, before we define our proposed threshold.



(a) (b)

Figure 4.1: (a) Comparison of similarity values of the terms *Book* and *Dwarfish* to 580K terms between models M and P. (b) Histogram of similarity values of an arbitrary term to all the other terms in the collection for 100, 200, 300, and 400 dimensions.

4.1.1 Uncertainty of Similarity

In this section, we make a series of practical observations on word embeddings and the similarities computed based on them.

To observe the uncertainty, let us consider two models P and M. To create each instance, we train word2vec SkipGram models similar to the way we did in Chapter 3 with the sub-sampling parameter of 10^{-5}, context windows of 5 terms, epochs of 25, and term count threshold 20 on the Wikipedia dump file for August 2015, after applying the Porter stemmer. Each model has a vocabulary of approximately 580k terms. They are identical in all ways except their random starting point.

Figure 4.1a shows the distances between two terms and all other terms in the dictionary, for the two models, in this case of dimensionality 200. For each term, we have approximately 580k points on the plot. As we can see, the difference between similarities calculated in the two models, appears (1) greater for low similarities, and (2) greater for a rare term (Dwarfish) than for a common term (Book). We can also observe that there are very few pairs of terms with very high similarities.

Let us now explore the effect of dimensionality on similarity values and a uncertainty. Before that, in order to generalize the observations to an arbitrary term, we had to consider a set of "representative" terms. What exactly "representative" means is of course debatable. We took 100 terms recently introduced in the query inventory method by Schnabel et al. [SLMJ15]. They claim that the selected terms are diverse in frequency and part of speech over the collection terms. In the remainder of the chapter, we refer to *arbitrary* term as an aggregation over the representative terms i.e. each value related to the arbitrary term is the average of the values of the representative terms.

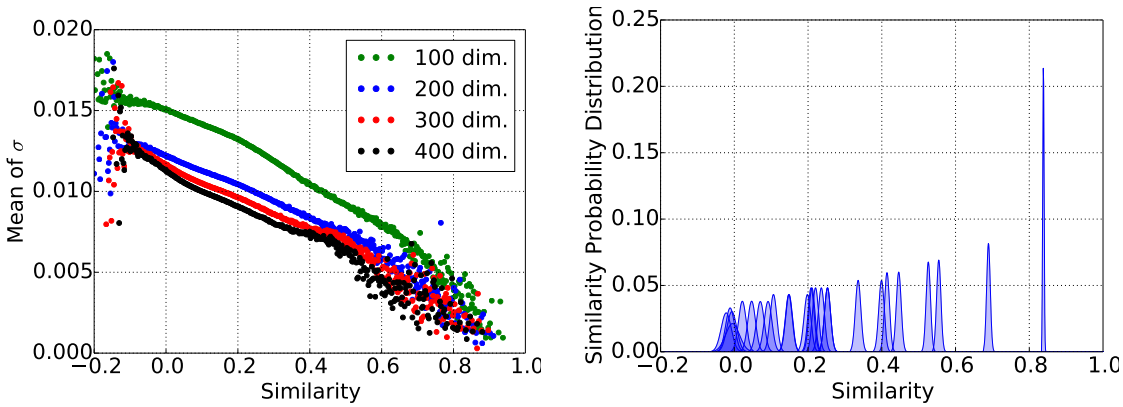


Figure 4.2: (a) Standard deviation for similarity values. Points are the average over similarity intervals with equal lengths of 2.4×10^{-4} (b) Probability distribution of similarity values for the term *Book* to some other terms.

Figure 4.1b shows the fitted curve to the histogram of similarity values for models of different dimensionalities. As we can see, similarities are in the $[-0.2, 1.0]$ range and have positive skewness (the right tail is longer). As the dimensionality of the model increases, the kurtosis also increases (the histogram has thinner tails).

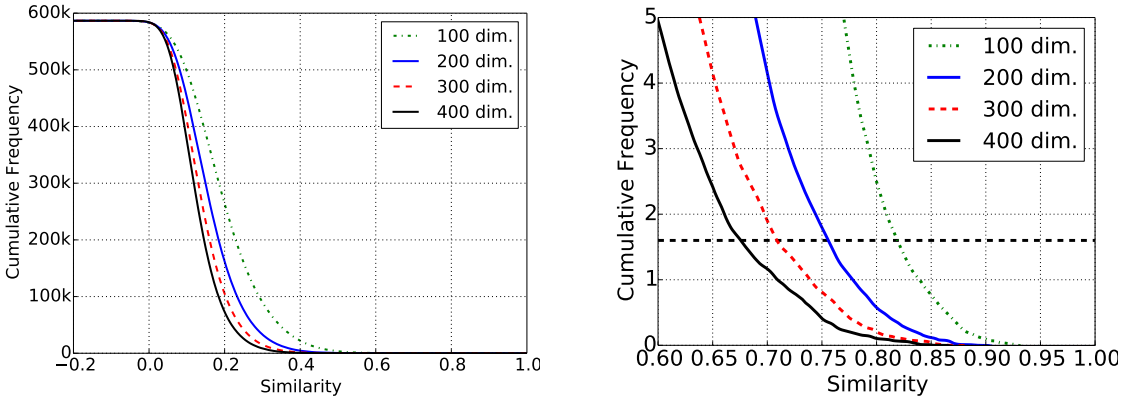
Let us first suggest a concrete definition for uncertainty: We quantify the uncertainty of the similarity between two terms as the standard deviation σ of similarity values obtained from a set of identical models. We refer to identical models as the models created using the same method, parameters, and corpus. However as shown before, the similarity values of each term pair in each model are slightly different. The uncertainty of similarity between the terms x and y is therefore formulated as follows:

$$\sigma_{x,y} = \sqrt{\frac{1}{|M|}\sum_{m\in M}(sim(\vec{x}_m - \vec{y}_m) - \mu)^2}, \quad \text{where} \quad \mu = \frac{\sum_{m\in M} sim(\vec{x}_m - \vec{y}_m)}{|M|} \tag{4.1}$$

where M is the set of identical models, $\vec{x}_m$ is the vector representation of term x in model m, and sim is a similarity function between two vectors.

To observe the changes in standard deviation, for every dimensionality, we create five identical SkipGram models ($|M| = 5$).

Figure 4.2a plots the standard deviation, against the similarity values, for different model dimensionalities. For the sake of clarity in visualization, we split the similarity values into 500 equal intervals (each 2.4×10^{-4}) and average the values in each interval. The plots are smooth in the middle and scattered on the head and tail as the majority of similarity values are in the middle area of the plots and therefore the average values are consistent. However, we can observe that overall, as the similarity increases, the standard deviation, i.e. the uncertainty, decreases.



(a) (b)

Figure 4.3: (a) Mixture of cumulative probability distributions of similarities in different dimensions (b) Expected number of neighbors around an arbitrary term with confidence interval. The average number of synonyms in WordNet (1.6) is shown by the dash-line.

We also observe a decrease in standard deviation as the dimensionality of the model increases. On the other hand, the differences between models decrease as the dimension increases such that the models of dimension 300 and 400 seem very similar in comparison to 100 and 200. The observation shows a probable convergence in the uncertainty at higher dimensionalities.

These observations show that the similarity between terms is not an exact value but can be considered as an approximation whose variation is dependent on the dimensionality and similarity range. We use the outcome of these observations in the following.

4.1.2 Continuous Distribution of Neighbors

We have demonstrated that the similarity values of a pair of terms obtained from identical embedding models are slightly different. In the absence of additional information, we assume that these similarity values follow a normal distribution.

To estimate this probability distribution, we use the mean and standard deviation values in Section 4.1.1. Figure 4.2b shows the probability distribution of similarities for term *Book* to 25 terms in different similarity ranges[1]. As observed before, by decreasing similarity, the standard deviation of the probability distributions increases.

We use these probability distributions to provide a representation of the expected number of neighbors around an arbitrary term in the spectrum of similarity values: We first calculate the Cumulative Distribution Functions (CDF) of the probability distributions. We then subtract the CDF values from 1 which only reverses the direction of the distributions (from increasing left-to-right on X-axis to right-to-left). Finally, we

[1] we do not plot all the terms in the model to maintain the readability of the plot

Table 4.1: Proposed thresholds for various dimensionalities

Dimensionality	Threshold Boundaries		
	Lower	**Main**	Upper
100	0.802	**0.818**	0.829
200	0.737	**0.756**	0.767
300	0.692	**0.708**	0.726
400	0.655	**0.675**	0.693

accumulate all the cumulative distribution functions by summing all the values, shown in Figure 4.3a. The values on this plot indicate the number of expected neighbors that have greater or equal similarity values to the term than the given similarity value. We can see the number of all the terms in the model (580k) in the lowest similarity value (-0.2) which then rapidly drops as the similarity increases. This representation of the expected number of neighbors in Figure 4.3a has two benefits: (1) the estimation is continuous and monotonic, and (2) it considers the effect of uncertainty based on five models.

As noted before, the notion of *arbitrary* term is in fact an average over the 100 representative terms. Therefore, in calculating the representation of the expected number of neighbors, we also consider the confidence interval around the mean. This interval is shown in Figure 4.3b. Here, the representation is zoomed on the lower right corner of Figure 4.3a. The shaded area around each line shows the confidence interval of the estimation.

This continuous representation is used in the following for defining the threshold for the semantically related terms.

4.1.3 Similarity threshold

Given the expected number of neighbors around the arbitrary term, represented in Figure 4.3a and Figure 4.3b, the question is "*what is the best threshold for filtering the related terms?*". In order to address the question, we hypothesize that since this general threshold tries to separate related from unrelated terms, it can be estimated from the average number of synonyms over the terms. Therefore, we transform the above question into a new question: "*What is the expected number of synonyms for a term in English?*"

To answer this, we exploit WordNet. We consider the distinct terms in the related synsets to a term as its synonyms, while filtering the terms containing multiple words (e.g. Natural Language Processing, shown in WordNet as Natural_Language_Processing form) since in creating the word embedding models such terms are considered as separated terms (one word per term). The average number of synonyms over all the 147306 terms of WordNet is 1.6, while the standard deviation is 3.1.

Using the average value of the synonyms in WordNet, we define our threshold for each model dimensionality as the point where the estimated number of neighbors in Figure 4.3b

is equal to 1.6. We also consider an upper and lower bound for this threshold based on the points on the similarity axis at which the confidence interval plots cross the horizontal line of the average value. The results are shown in Table 4.1.

In the following sections, we validate the hypothesis by evaluating the performance of the proposed thresholds with an extensive set of experiments.

4.2 Experiment Setup

We test the effectiveness of our threshold in an Ad-hoc retrieval task on IR test collections by evaluating the results of applying various thresholds to retrieve the related terms.

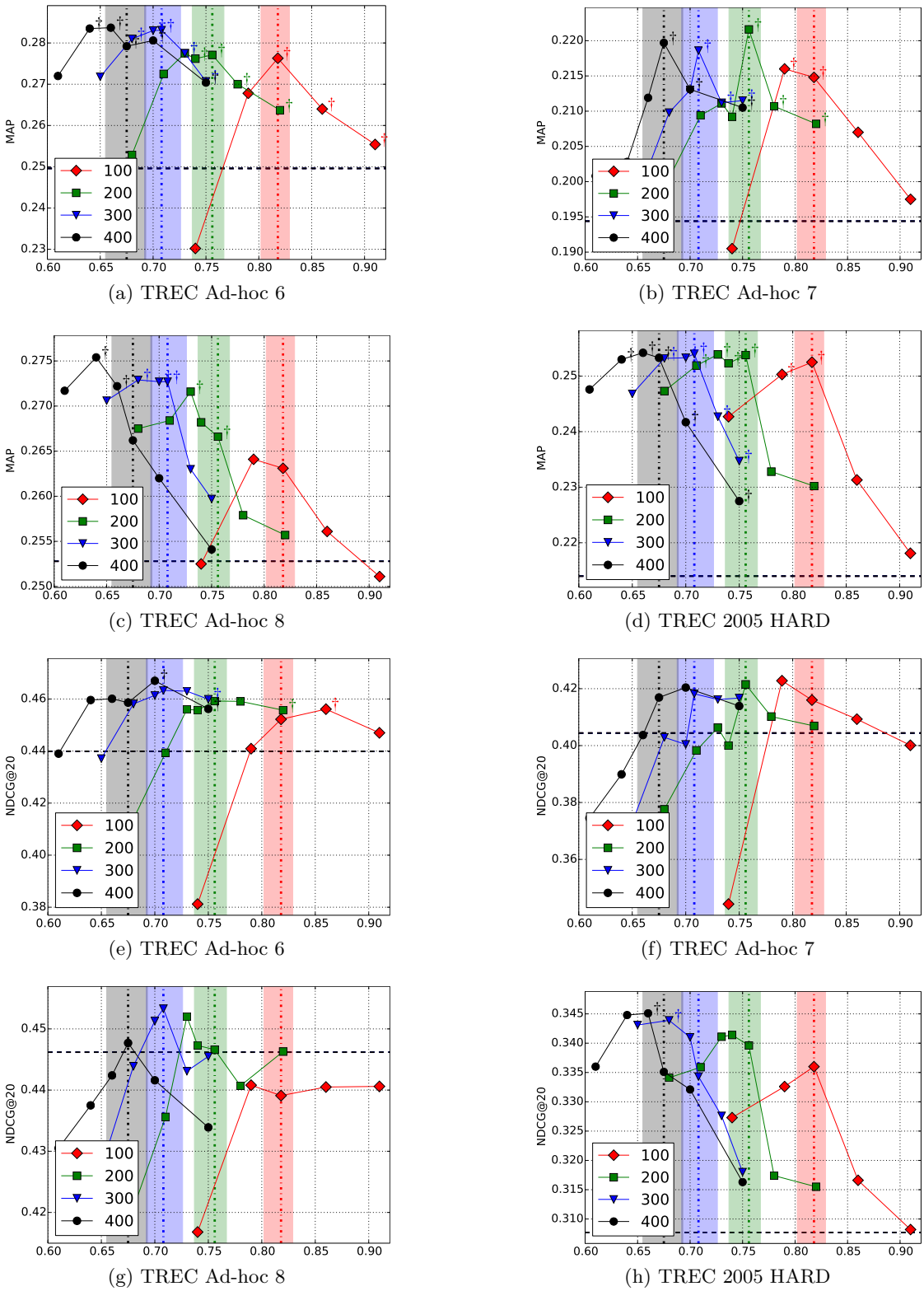
We use two relevance scoring approaches: query language model [PC98] and BM25 methods as two widely used and established methods in IR. As mentioned before, to exploit the set of related terms provided by word embeddings, we use the Generalized Translation Model and Extended Translation Model for BM25 and language modeling (Section 3.1.2).

We evaluate our approach on four test collections: TREC 6, TREC 7, and TREC 8 of the Ad-hoc track, and TREC 2005 HARD track (statistics available in Table 3.1). For pre-processing, we apply the Porter stemmer and remove stop words using a small list of 127 common English terms.

In order to compare the performance of the thresholds, we test a variety of threshold values for each model. The thresholds cover a set of values on both sides of our introduced thresholds: for 100 dimension {0.67, 0.70, 0.74, 0.79, 0.81, 0.86, 0.91, 0.94, 0.96}, 200 dimension {0.63, 0.68, 0.71, 0.73, 0.74, 0.76, 0.78, 0.82}, 300 dimension {0.55, 0.60, 0.65, 0.68, 0.70, 0.71, 0.73, 0.75}, and 400 dimension {0.41, 0.54, 0.61, 0.64, 0.66, 0.68, 0.70, 0.71, 0.75}.

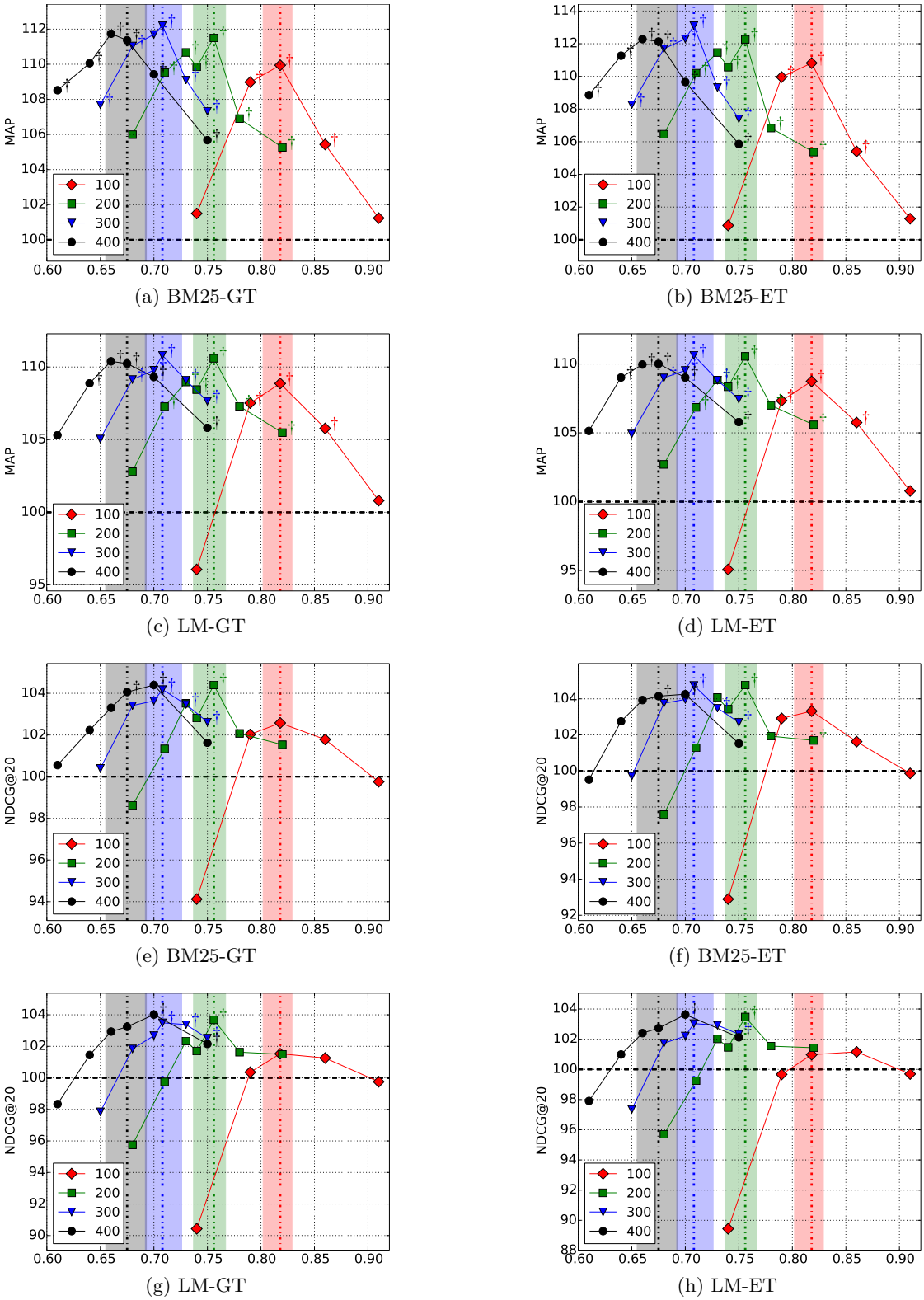
We set the basic models (language model or BM25) as baseline and test the statistical significance of the improvement of the translation models with respect to their basic models (indicated by the symbol †). Since the parameter μ for Dirichlet smoothing of the translation language model and also b, k_1, and k_3 for BM25 are shared between the methods, the choice of these parameters is not explored as part of this study and we use the same set of values as the previous chapter. The statistical significance test are done using the two sided paired t-test and statistical significance is reported for $p < 0.05$.

The evaluation of retrieval effectiveness is done with respect to Mean Average Precesion (MAP) and Normalized Discounted Cumulative Gain at cut-off 20 (NDCG@20), as standard measures in Ad-hoc information retrieval. Similar to Chapter 3, we consider MAP and NDCG over the condensed lists [Sak07].



(a) TREC Ad-hoc 6
(b) TREC Ad-hoc 7
(c) TREC Ad-hoc 8
(d) TREC 2005 HARD
(e) TREC Ad-hoc 6
(f) TREC Ad-hoc 7
(g) TREC Ad-hoc 8
(h) TREC 2005 HARD

Figure 4.4: MAP (above) and NDCG@20 (below) evaluation of the BM25 Extended Translation model on TREC 6, TREC 7, TREC 8 Ad-hoc, and TREC 2005 HARD for different thresholds (X-axes) and word embedding dimensions. Significance is shown by †. Vertical lines indicate our thresholds and their boundaries in different dimensions. The baseline is shown by the horizontal line. To maintain visibility, points with very low performance are not plotted.



(a) BM25-GT (b) BM25-ET

(c) LM-GT (d) LM-ET

(e) BM25-GT (f) BM25-ET

(g) LM-GT (h) LM-ET

Figure 4.5: Percentage of improvement of the relevance scoring models BM25 and Language Model (LM), combined with the Generalized Translation (GT) and Extended Translation (ET) models with respect to the baselines (standard LM and BM25) with the MAP (above) and NDCG@20 (below) evaluation measures for different thresholds, and word embedding dimensions, aggregated over all the collections.

Table 4.2: Examples of similar terms, selected with our threshold

book: publish, republish, foreword, reprint, essay
eagerness: hoping, anxious, eagerness, willing,wanting
novel: fiction, novelist, novellas, trilogy
microbiologist: biochemist, bacteriologist, virologist
shame: ashamed
guilt: remorse
Einstein: relativity
estimate, dwarfish, antagonize: no neighbors

4.3 Results and Discussion

The evaluation results of the MAP and NDCG@20 measures of the BM25 Extended Translation (BM-ET) model on the four test collections, with vectors in 100, 200, 300, and 400 dimensions are shown in Figure 4.4. We only show the detailed results of the BM-ET model as it has shown the best overall performance among the other translation models in the previous chapter. For each dimension, our threshold and its boundaries (the interval between the lower and upper bound in Table 4.1) are shown with vertical lines. The baseline (basic BM25) is shown as the horizontal line. Significant differences of the results to the baseline are marked by the † symbol.

The plots show that the performance of the translation models are highly dependent on the choice of the threshold value. In general, we can see a trend in all the models: the results tend to improve until reaching a peak (optimal threshold) and then converge to the baseline. Based on this general behavior, we can assume that including the terms whose similarity values are less than the optimal threshold introduces noise and deteriorates the results while using the cutting point greater than the optimal threshold filters the related terms too strictly[2]. We test the statistical significance of the differences between the results of the optimal and proposed threshold in all the experiments (both evaluation measures, all relevance scoring models, collections, and dimensions), observing no significant difference in any of the cases.

In order to have an overview of the improvements for all the models, we calculate the gain of each relevance scoring model for different thresholds and dimensionalities over its corresponding baseline and average the gains on the four collections. The scoring models are BM25 and Language Model (LM), combined with the Generalized Translation (GT) and Extended Translation (ET) models. The results for MAP and NDCG are depicted in Figure 4.5. In all the translation models, our proposed threshold is optimal for dimensions 100, 200, and 300. In dimension 400, the significance test between the best results and the results achieved from our threshold does not show any significant difference. This justifies the choice of the proposed threshold as a generally stable and effective cutting-point for identifying related terms.

To observe the effect of the proposed threshold, let us take a closer look at the terms,

[2]A more in-depth analysis is provided in Section 5.2

filtered as related terms. Table 4.2 shows some examples of the retrieved terms when using the word embedding model with 300 dimensions with our threshold (same as optimal in this dimension for all the translation models). As expected, the examples show the strong differences in the number of similar terms for various terms. The mean and standard deviation of the number of similar terms for all the query terms of the tasks is 1.5 and 3.0 respectively. Almost half of the terms are not expanded at all. We can observe the similarity between this calculated mean and standard deviation and the aggregated number of synonyms we observed in WordNet in Section 4.1.3—mean of 1.6 and standard deviation of 3.1. It appears that although the two semantic resources (WordNet and word2vec) cast the notion of similarity in different ways and their provided sets of similar terms are different, their values of mean and standard deviation for the number of similar terms are very close.

4.4 Summary

In this chapter, we analytically explore the thresholds on similarity values of word embedding to select related terms. Based on empirical observations on various models generated by different instances of an identical model, we estimate the variance of the cosine similarity value between two term vectors, allowing practical use of similarity values. The proposed threshold is estimated based on a novel representation of the neighbors around an arbitrary term, taking into account the empirically-measured variance of similarity values.

We extensively evaluate the application of the introduced threshold on four information retrieval collections using four relevance scoring models. The results show that the proposed threshold is identical to the optimal threshold (obtained by parameter scan) in the sense that its results on Ad-hoc retrieval tasks are either equal to or statistically indistinguishable from the optimal results.

CHAPTER 5

Fusion of Semantic Models with Window and Document Contexts

The effective model for choosing related terms to enrich queries has been explored for decades in information retrieval literature and approached using a variety of data resources. Early studies explore the use of collection statistics. They identify the global context of two terms either by directly measuring term co-occurrence in a context (i.e. document) [PW91] or after applying matrix factorization as in the LSI method [DDF+90]. Later studies show the higher effectiveness of local approaches (i.e. using pseudo-relevant documents) [XC96]. As shown in Chapter 3 and also other recent studies [ZC16, ZC17], approaches to exploit word embedding for IR are not only competitive to the local approaches but also that combining the approaches brings further improvements in comparison to each of them alone.

Word embedding methods capture the co-occurrence relations between the terms, based on an approximation of the likelihood of their appearances in similar window-contexts. However, since the concept of term relatedness is defined as a similarity proximity between such vector representations, some related terms might not fit to the retrieval needs and eventually deteriorate the results. For instance, antonyms ('cheap' and 'expensive') or co-hyponyms ('schizophrenia' and 'alzheimer', 'mathematics' and 'physics', countries, months) share common window-contexts and are therefore considered as related in the word embedding space, but can potentially bias the query to other topics.

In this chapter, we address this problem by studying the effect of similarity achieved from global (document) context as a complement to the window-context based similarity. In fact, similar to the earlier studies [PW91, SM83], we assume each document to be a coherent information unit and consider the co-occurrence of terms in documents as a means of measuring their topical relatedness. Based on this assumption, we hypothesize that to mitigate the problem of topic shifting, the terms with high word embedding

Table 5.1: Test collections used in this chapter

Name	Collection	# Queries	# Documents
TREC Ad-hoc 1&2&3	Disc1&2	150	740449
TREC Ad-hoc 6&7&8	Disc4&5	150	556028
HARD	AQUAINT	50	1033461

similarities also need to share similar global contexts. In other words, if two terms appear in many similar window-contexts, but they share few document-contexts, they probably reflect different topics and should be removed from the related terms.

To examine this hypothesis, we analyze the effectiveness of each related term, when added to the query. Our approach is similar to that of Cao et al. [CNGR08] on pseudo-relevance feedback. Our analysis shows that the set of related terms from word embedding has a high potential to improve state-of-the-art retrieval models. Based on this motivating observation, we explore the effectiveness of using word embedding's similar term when filtered by global context similarity. Our evaluation on three test collections shows the importance of using global context, as combining both the similarities significantly improves the results.

The chapter is organized as follows: Section 5.1 analyses the effectiveness of the extended terms, followed by describing our post-filtering methods in Section 5.2. We discuss the retrieval results in Section 5.3 and finally conclude the work in Section 5.4.

5.1 Preliminary Analysis

In this section, we investigate the retrieval effectiveness of the similar terms to the query terms, when incorporated in a retrieval model. We set up our experiment similar to the previous chapters. Since TREC Ad-hoc 6, 7, and 8 share the same collection of documents, to have a possible higher number of queries per collection required for cross-validation in the next section, we merge these there collections into one (TREC 678). We therefore conduct the experiments on three test collections, shown in Table 5.1. For word embedding vectors, we train the word2vec SkipGram model with 300 dimensions and the same parameters in the previous two chapters on the Wikipedia dump file for August 2015. We use the Porter stemmer for the Wikipedia corpus as well as retrieval. Similar to Chapter 4, we use the novel BM25 translation model and translation Language Modeling (Chapter 3) for retrieval, denoted as $\widehat{BM25}$ and $\widehat{LM}$. We specifically focus on the Extended Translation models as they have shown relatively better performance than the Generalized Translation models in the previous experiments. To explore the effectiveness of less similar terms, we consider the threshold values of {0.60, 0.65..., 0.80}, used for filtering the related terms of the word embedding model. However, for the final results we only consider threshold of 0.7 as introduced in Chapter 4.

Table 5.2: The percentage of the good, bad and neutral terms. #Rel averages the number of related terms per query term.

Collection	Threshold 0.60				Threshold 0.80			
	#Rel	Good	Neutral	Bad	#Rel	Good	Neutral	Bad
TREC 123	8.2	7%	84%	9%	1.3	19%	68%	13%
TREC 678	8.8	9%	78%	14%	1.2	34%	48%	18%
HARD	10.3	8%	77%	15%	1.1	39%	44%	17%
ALL	8.1	8%	81%	11%	1.2	27%	58%	15%

The rests of the experiment settings are the same as the previous chapters: The parameters μ for Dirichlet prior of the Language Modeling (LM), b, k_1, and k_3 for $BM25$ are set to the same values as in Chapter 3. The statistical significance tests are done using the two sided paired t-test and significance is reported for $p < 0.05$, and finally, the evaluation of retrieval effectiveness is done with respect to NDCG at 20 and MAP both over the condensed list.

We start with an observation on the effectiveness of each individual related term. To measure it, we use the $\widehat{LM}$ model as in Chapter 3 it has shown slightly better results than the $\widehat{BM25}$ model. Similar to Cao et al. [CNGR08], given each query, for all its corresponding related terms, we repeat the evaluation of the IR models where each time the set of related terms $R(t)$ (Section 3.1.1) consists of only one of the related terms. For each term, we calculate the differences between its Average Precision (AP) evaluation result and the result of the original query and refer to this value as the *retrieval gain* or *retrieval loss* of the related term.

Similar to Cao et al. [CNGR08], we define *good*/*bad* groups as the terms with retrieval gain/loss of more than 0.005 for the AP measure, and assume the rest with smaller gain or loss values than 0.005 as *neutral* terms. Table 5.2 summarizes the percentage of each group for the lowest (0.6) and highest (0.8) threshold. The average number of related terms per query term is shown in the `#Rel` field. As expected, the percentage of the good terms is higher for the larger threshold, however—similar to the observation on pseudo-relevance feedback [CNGR08]—most of the expanded terms (58% to 81%) have no significant effect on performance.

Let us imagine that we had a priori knowledge about the effectiveness of each related term and were able to filter terms with negative effect on retrieval. We call this approach `Oracle` post-filtering as it shows us the maximum performance of each retrieval model. Based on the achieved results, we provide an approximation of this approach by filtering the terms with retrieval loss.

Figure 5.1 shows the percentage of relative MAP and NDCG improvement of the $\widehat{LM}$ model with and without post-filtering with respect to the original LM model[1]. The

[1]the $\widehat{BM25}$ models show similar trend and are depicted later in the chapter in Figure 5.4b

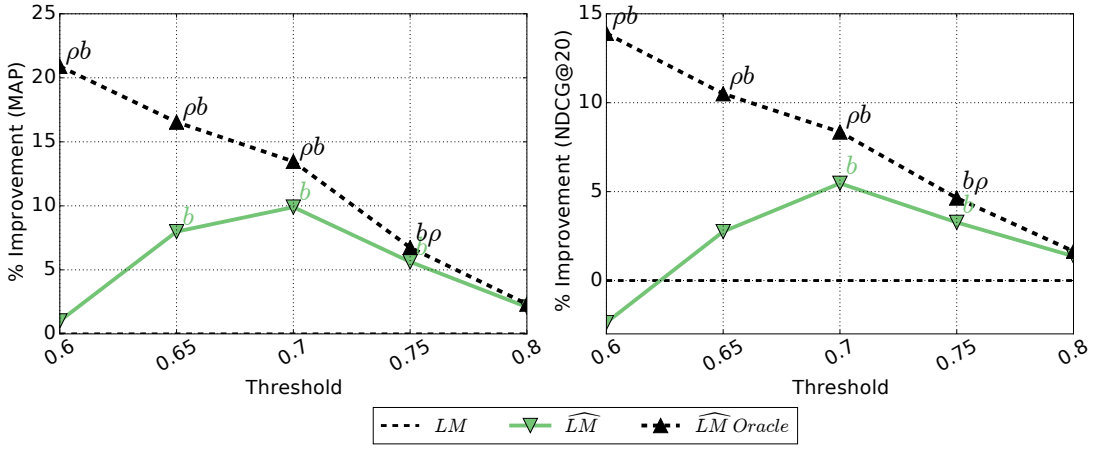


Figure 5.1: The percentage of relative improvement of $\widehat{LM}$ models to the basic LM, aggregated over all the collections for MAP and NDCG@20 measures. The b and ρ signs show the significance of the improvement to the basic models and the extended models without post filtering respectively.

results are aggregated over the three collections. For each threshold the statistical significance of the improvement with respect to two baselines are computed: (1) against the basic LM, shown with the b sign and (2) against the translation models without post filtering, shown with the ρ sign.

As observed in Chapter 4, for the thresholds less than 0.7 the retrieval performance of the translation models (without post filtering) decreases as the added terms introduce more noise. However, the models with the `Oracle` post filtering continue to improve over the baselines further for the lower thresholds with high margin. These demonstrate the high potential of using related terms from word embedding but also show the need to customize the set of terms for IR. We propose an approach to this customization using the global-context of the terms in the following.

5.2 Global-Context Post Filtering

Looking at some samples of retrieval loss, we can observe many cases of topic shifting: e.g. Latvia as query term is expanded with Estonia, Ammoniac with Hydrogen, Boeing with Airbus, and Alzheimer with Parkinson. As mentioned before, our hypothesis is that for the terms with high window-context similarity (i.e. word2vec similarity) when they have high global context similarity (i.e. co-occurrence in common documents), they more probably refer to a similar topic (e.g. USSR and Soviet) and with low global context similarity to different topics (e.g. Argentina and Nicaragua).

To capture the global context similarities, we examine several term semantic similarity methods based on document context. In the following, we explain each method.

Our first set of measures, applied in some older studies [PW91], use the document set of a term, namely the set of documents that term t appears in, denoted as DS_t. Using the document sets of query term q and an extended term t (DS_q and DS_t respectively), we define the Dice, Jaccard, and PMI (denoted as PMI^{DS} to avoid confusion with Eq. 2.2) global similarity measures as follows:

$$\text{Dice}(q,t) = \frac{2\,|DS_q \cap DS_t|}{|DS_q|\,|DS_t|} \tag{5.1}$$

$$\text{Jaccard}(q,t) = \frac{|DS_q \cap DS_t|}{|DS_q \cup DS_t|} \tag{5.2}$$

$$\text{PMI}^{\text{DS}}(q,t) = \log\left(\frac{p(DS_q \cap DS_t)}{p(DS_q)p(DS_t)}\right) \tag{5.3}$$

where $p(DS) = |DS| \,/\, |D|$, and $|D|$ is the number of documents in the collection.

The second set of measures first define a vector representation of a term based on its document context, and then compute the similarity between two term vectors using the Cosine function. These methods are conceptually the same as some of the word representation models explained in Section 2.1, but they use document context instead of window context. The first two methods consider term vectors with dimensionality of the number of documents in the collection (explicit representation), with weights given either as simple incidence (i.e. 0/1), or by TFIDF. The third method applies Singular Value Decomposition on the TFIDF weighted term-document matrix, resulting in the Latent Semantic Indexing (LSI) method [DDF$^+$90], described in Section 2.1.2. Similar to the word2vec vectors, we create the LSI vectors with 300 dimensions.

Finally, we compute these measures using the statistics of each collection as well as the Wikipedia collection. This results in 12 sets of similarities (Dice, Jaccard, PMI^{DS}, Incidence Vectors, TFIDF Vectors, LSI Vectors)$\times$(collection, Wikipedia). We refer to these similarity value lists as global context features.

Let us first observe the relationship between one of these feature, namely LSI when using collection statistics, and word2vec similarities. Figure 5.2 plots the retrieval gain/loss of the terms of all the collections based on their word2vec similarities as well as LSI. The size of each circle shows the amount of gain (green) or loss (red) in the retrieval performance by using a term. For clarity, we only show the terms with the retrieval gain/loss of more than 0.01. The area with high word2vec and LSI similarity (top-right) contains most of the terms with retrieval gain. On the other hand, regardless of the word2vec similarity, the area with lower LSI tend to contain relatively more cases of retrieval loss. This observation encourages the exploration of a set of thresholds for global context features to post filter the terms retrieved by embedding.

To find the thresholds for global context features, we explore the highest amount of total retrieval gain after filtering the related terms with similarities higher than the thresholds.

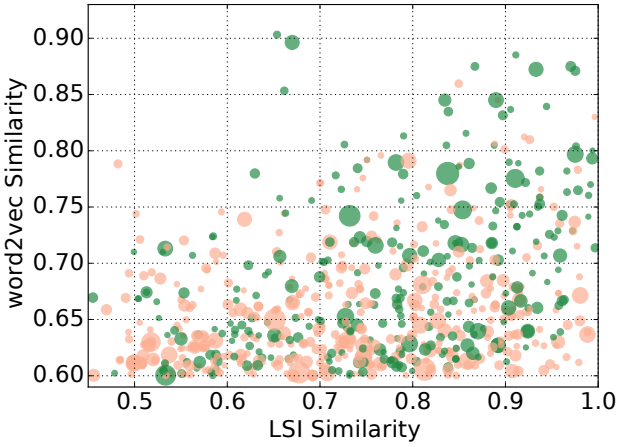


Figure 5.2: Retrieval gain or loss of the related terms for all the collection. The red (light) color indicate retrieval loss and the green (dark) retrieval gain.

We formulate it by the following optimization problem:

$$\underset{\Theta}{\operatorname{argmin}} \sum_{i=1}^{N} 1 \left[\bigcap_{j=1}^{F} x_j > \theta_j \right] g_i \tag{5.4}$$

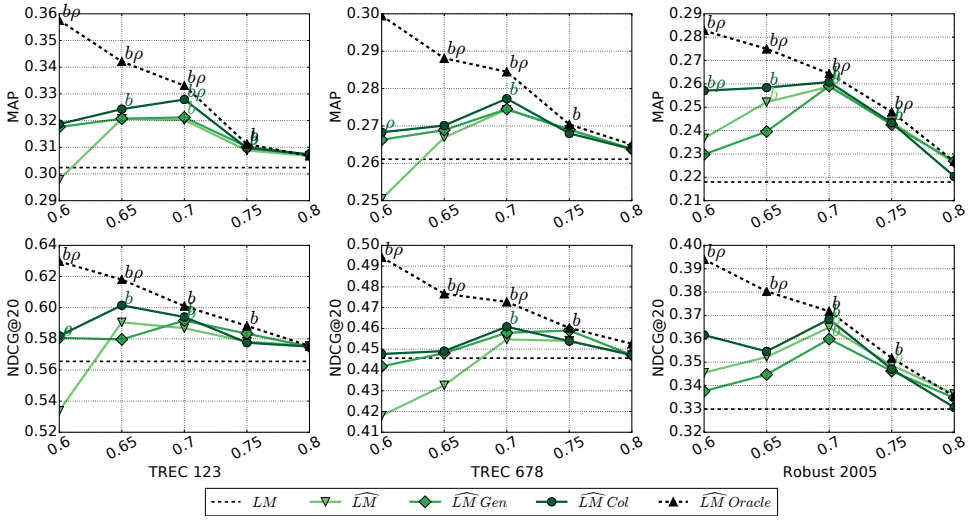
where 1 is the indicator function, N and F are the number of terms and features respectively, Θ indicates the set of thresholds θ_j, x_j the value of the features, and finally g refers to the retrieval gain/loss.

We consider two approaches to selecting the datasets used to find the optimum thresholds: *per collection*, and *general*. In the per collection scenario (`Col`), for each collection we find different thresholds for the features. We apply 5-fold cross validation (folds are formed randomly) by first using the terms of the training topics to find the thresholds (solving Eq. 5.4) and then applying the thresholds to post filter the terms of the test topics. To avoid overfitting, we use the bagging method by 40 times bootstrap sampling (random sampling with replacement) and aggregate the achieved thresholds.

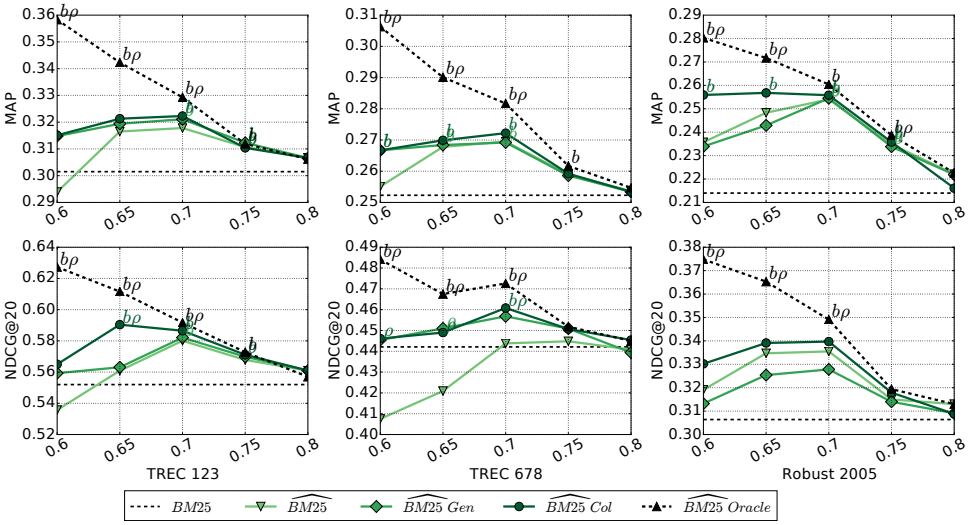
In the general approach (`Gen`), we are interested in finding a 'global' threshold for each feature, which is independent of the collections. As in this approach the thresholds are not specific to each individual collection, we use all the topics of all the test collections to solve the optimization problem.

5.3 Results and Discussion

To find the most effective set of features, we test all combinations of the 12 discussed features using the per collection (`Col`) post-filtering approach. Given the post-filtered terms with each feature set, we evaluate the $\widehat{LM}$ and $\widehat{BM25}$ models. Our results show that by only using the combination of LSI and TFIDF features defined on the collections



(a) LM



(b) $BM25$

Figure 5.3: Evaluation results of the LM and $BM25$ Extended Translation models with/without post filtering for MAP and NDCG@20 measures. The b and ρ signs show the significance of the improvement to $LM/BM25$ and $\widehat{LM}/\widehat{BM25}$ without post filtering respectively.

statistics, we achieve the best performance among various combinations, and adding any of the other features (containing the ones based on Wikipedia) does not bring any improvement. Therefore in the following, to simplify our computations for post filtering, we only use the combination of LSI and TFIDF similarity features with both using the collections statistics.

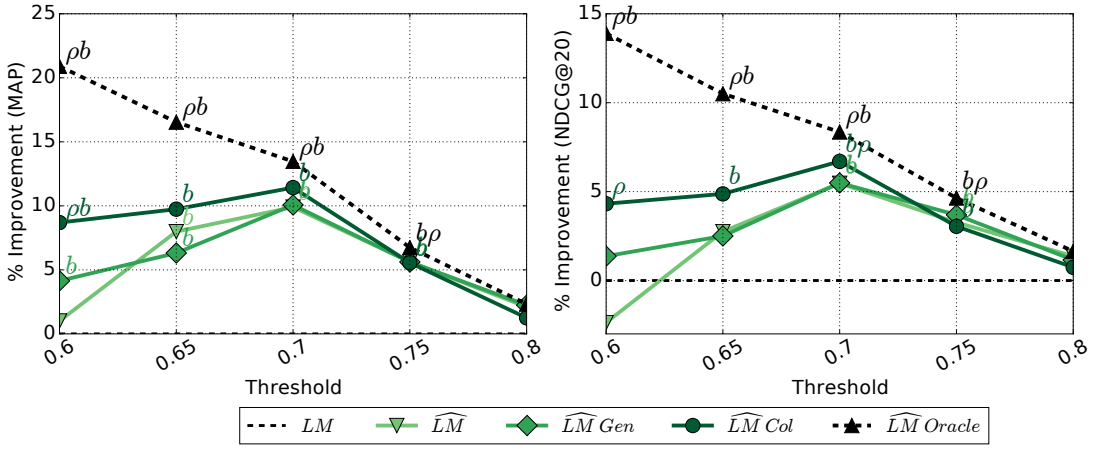
Table 5.3: MAP and NDCG@20 of the Extended Translation models (ET) when terms are filtered with word embedding threshold of 0.7 and post filtered with the `Gen` and `Col` approach, using the LSI and TFIDF features.

Collection	Method	MAP		NDCG@20	
		LM	**BM25**	**LM**	**BM25**
TREC 123	Basic	0.302	0.301	0.566	0.552
	ET	0.320	0.318	0.587	0.580
	ET+Gen	0.321 b	0.321 b	0.592 b	0.582 b
	ET+Col	0.328 $b\rho$	0.322 b	0.594 b	0.587 b
TREC 678	Basic	0.261	0.252	0.446	0.442
	ET	0.274	0.270	0.455	0.444
	ET+Gen	0.275	0.269 b	0.458	0.457
	ET+Col	0.277 b	0.272 b	0.461 b	0.461 $b\rho$
HARD	Basic	0.218	0.214	0.330	0.306
	ET	0.259 b	0.254 b	0.365 b	0.336
	ET+Gen	0.259 b	0.255 b	0.360 b	0.328
	ET+Col	0.261 b	0.256 b	0.368 b	0.340

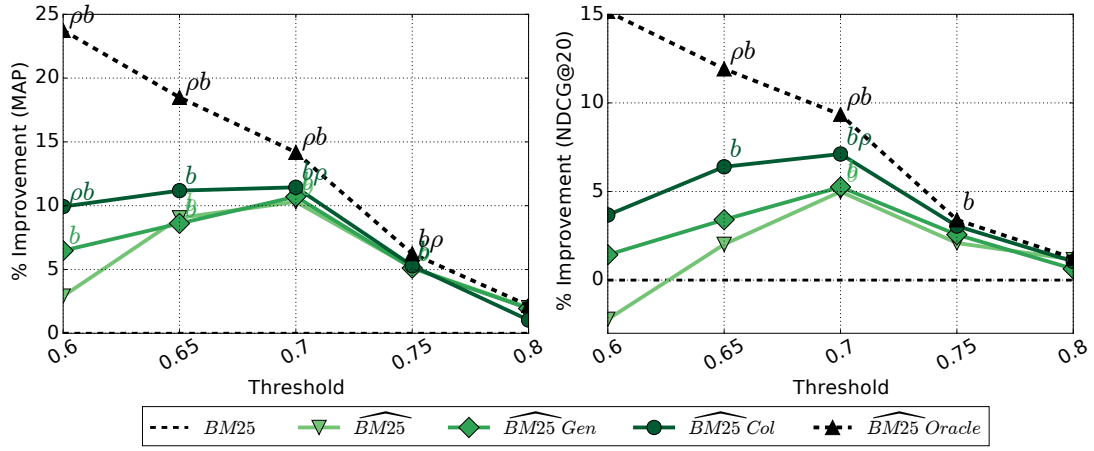
Figure 5.3 shows the evaluation results of the original Extended Translation models ($\widehat{LM}$ and $\widehat{BM25}$) and the Extended Translation models with post filtering based on the general (`Gen`) and per collection (`Col`) approaches. As before, statistical significance against the basic models is indicated by b and against the translation models without post filtering by ρ.

The results for both evaluation measures show the improvement of the Extended Translation models with post-filtering in comparison with the original ones. The models with post-filtering approaches specifically improve in lower word embedding thresholds, however similar to the original Extended Translation models, in average the best performance is achieved on word embedding threshold of 0.7. The results with word embedding threshold of 0.7 are summarized in Table 5.3. Comparing the post-filtering approaches, `Col` shows better performance than `Gen` as with the proposed word embedding threshold, it achieves significant improvements over both baselines in two of the collections.

Let us look again to the relative improvements, aggregated over the collections, shown in Figure 5.4. The figure is similar to Figure 5.1 but contains the results of the post filtering approaches. As shown, the `Col` approach shows better results than other models on both evaluation measures and IR models, and except in LM with MAP, the `Col` approach achieves significant improvement over both baselines.



(a) LM



(b) $BM25$

Figure 5.4: The percentage of relative improvement to the basic models, aggregated over all the collections for MAP and NDCG@20 measures. The plot is similar to Figure 5.1 but contains the results of `Col` and `Gen` approaches.

5.4 Summary

Word embedding methods use (small) window-context of the terms to provide dense vector representations, used to approximate term relatedness. In this chapter, we study the effectiveness of related terms, identified by both window-based and global (document) contexts, in document retrieval. We use two state-of-the-art translation models to integrate word embedding information for retrieval.

Our analysis shows a great potential to improve retrieval performance, damaged however by topic shifting. To address it, we propose the use of global context similarity, i.e. the co-occurrence of terms in larger contexts such as entire documents. Among various methods

to measure global context, we identify the combination of LSI and TFIDF as the most effective feature set in eliminating related terms that lead to topic shifting. Evaluating the IR models using two post-filtering approaches shows a significant improvement in comparison with the basic models as well as the Extended Translation models with no post-filtering. The results demonstrate the importance of global context for selecting the related term when combined with the window-context similarities.

CHAPTER 6

Interpretability in Word Embedding

Word embedding models provide significant benefit to information processing tasks. While easy to construct based on raw unannotated corpora, these dense representations and their estimation of term-term relatedness remain difficult to interpret and hard to analyze. In fact, when using word embedding, it remains opaque what the dimensions of the vectors refer to, or to what extend a semantic concept is present in the vector representation of a term.

As discussed in Section 2.1.1, a natural solution to this problem is using explicit representations of words i.e. vectors with clearly-defined dimensions, where each dimension represents an explicit concept such as a term, window of terms, or document. Such an explicit vector of a word is easily interpretable, as each dimension stands for the degree of relation between the word and the corresponding concept.

We also discussed the effectiveness of the explicit representations in Section 2.1.5. As shown by Levy et al. [LGD15], the recent explicit representation models such as Shifted Positive Point Mutual Information (SPPMI), explained in Eq. 2.10, show competitive performance in comparison to the state-of-the-art word embeddings on a set of term association test collections. Regarding efficiency, the explicit representations often require much bigger memory space in comparison to the low-dimensional dense vectors. However, in practice the memory issue can be mitigated by suitable data structures if the vectors are highly sparse.

As an alternative approach to improve interpretability, some recent work [FTY+15, SGL+16] propose methods to increase the sparsity of the dense vectors. The rationale of these approaches is that by having more sparse vectors, it becomes more clear which dimension of the vectors might be referring to which concepts in language.

In contrast to this approach, our first contribution in this chapter is in line with previous studies [LG14, LGD15] on providing fully interpretable vectors by proposing a novel explicit representation for the word2vec SkipGram model. We propose a method to transfer the low-dimensional (dense) vectors of a trained SkipGram model to explicit vector representations in a high-dimensional space. Our approach is in the opposite direction to the methods such as LSI or GloVe (discussed in Section 2.1.2), where they start from a high-dimensional matrix and result in low-dimensional embeddings. In contrast, the main objective of our work is to provide an interpretable variation of the SkipGram vectors, enabling error resolution and better causal analysis.

We evaluate our explicit SkipGram model on 6 term-to-term association benchmarks, showing results on par with the SPPMI model as the state of the art of explicit representation vectors. These results support the reliability of our approach to create high quality interpretable vectors of the SkipGram model.

To show an application of our explicit SkipGram representation, in our next contribution, we propose a novel approach based on explicit vectors to quantify the degree of gender bias in a corpus. We particularly focus on the inclination of a set of gender-neutral occupations to male or female in a Wikipedia English corpus.

As a study close to our work, Bolukbasi et al. [BCZ+16] quantify the gender bias of an occupation by calculating the semantic similarity of the vectors of the terms 'she' and 'he' (V_{she} and V_{he}), as the representative of female and male, to the vector of the occupation using the SkipGram model. We point out an intrinsic issue in this approach, by arguing that V_{she} and V_{he} are not precise representatives of female and male concepts, since due to bias in language they also contain other types of concepts, specially the ones related to occupations. For instance, if 'nurse' is biased to female, we expect that V_{nurse} contains many concepts related to female. However, it also means that V_{she} contains high relation to the concept 'nurse'. We refer to this characteristic of word embedding as *circularity*. Considering this trait, given that V_{nurse} naturally contains the concept 'nurse', calculating the semantic similarity between V_{she} and V_{nurse} (as the degree of bias of 'nurse' to female) is wrongly inclined by the 'nurse' concept.

To address the issue caused by circularity, we exploit the interpretability characteristic of the explicit SkipGram representations, by selecting only the gender-related concepts (dimensions) of the gender vectors. In our approach, the bias towards female is quantified by defining a new gender vector V_{SHE}, where its female-related dimensions are explicitly set to the ones of V_{she} and the rest to zero (the same process for bias towards male by defining the vector V_{HE}).

The proposed gender vectors V_{SHE} and V_{HE} therefore only consist of gender-specific concepts which arguably provide a more precise approach to gender bias quantification. These results specially demonstrate the high bias of some specific jobs to female-specific concepts. This inherent bias in data and therefore word representations can potentially be propagated to information systems, leading to biased decisions.

In the following, first we introduce our novel explicit representation model in Section 6.1,

followed by evaluating the performance of the introduced representation. We then discuss our approach to quantifying the degree of gender bias in Wikipedia using the explicit SkipGram representation and report the results in Section 6.2. Finally, we conclude the chapter in Section 6.3.

6.1 Explicit SkipGram

In this section, we first explain our approaches to create explicit representations of the SkipGram model, followed by evaluation and comparison of the proposed representations.

6.1.1 Definition

To define our novel explicit representations, let us first revisit the $p(y = 1|w, c)$ probability in the word2vec SkipGram model (referred to as SG in the rest of this chapter), explained in Section 2.1.4 and Eq. 2.8. $p(y = 1|w, c)$ measures the probability that the co-occurrence of two terms w and c comes from the training corpus and not from a random distribution. The purpose of this probability is in fact related to the conceptual goal of the PMI-based representations i.e. to distinguish a genuine from a random co-occurrence (Section 2.1.1). Indeed, both of these probabilities aim to capture the first-order relationship between two terms, based on the corpus at hand. Based on this idea, an immediate way of defining an explicit representation would be to use Eq. 2.8 as follows:

$$ExpSG^c(w) = p(y = 1|w, c) = \sigma(V_w \tilde{V}_c) \tag{6.1}$$

where as in Section 2.1, V and $\widetilde{V}$ are the set of term and context vectors, and σ is the sigmoid function.

This *Explicit SkipGram (ExpSG)* representation assigns a value between 0 to 1 to the first-order relation of each pair of terms. It is however intuitive to consider that the very low values do not represent a genuine relation and can potentially introduce noise in computation. Such very low values can be seen in the relation of a term to very frequent or completely unrelated terms. We can extend this idea to all the values of ExpSG, i.e. some portion (or all) of every relation contains noise.

To measure the noise in ExpSG, we use the definition of noise probabilities in the Negative Sampling approach: the expectation value of $p(y = 1|w, c)$ where c (or w) is randomly sampled from the dictionary for several times. Based on this idea, we define the *Reduced Explicit SkipGram (RExpSG)* model by subtracting the two expectation values from ExpSG:

$$RExpSG^c(w) = ExpSG^c(w) - \mathop{\mathbb{E}}_{\check{c}\sim\mathcal{N}} p(y = 1|w, \check{c}) - \mathop{\mathbb{E}}_{\check{w}\sim\mathcal{N}} p(y = 1|\check{w}, c) \tag{6.2}$$

where $\mathbb{E}$ is the expectation value over any $\check{c}$ term, sampled from the noisy distribution $\mathcal{N}$.

Since the expectation values can be calculated off-line, in contrast to Negative Sampling (restricted to a set of k sampled terms), we compute it over the entire vocabulary:

$$\mathop{\mathbb{E}}_{\check{w}\sim\mathcal{N}} p(y=1|\check{w},c) = \frac{\sum_{i=1}^{|W|} \#(\check{w}_i) \cdot \sigma(V_{\check{w}_i}\widetilde{V}_c)}{\sum_{i=1}^{|W|} \#(\check{w}_i)} \tag{6.3}$$

For the sampling of the context term $\check{c}$, similar to SG and PMI_α (Eq. 2.11), we apply the `cds` method (Section 2.1.4) by raising frequency to the power of α, as follows:

$$\mathop{\mathbb{E}}_{\check{c}\sim\mathcal{N}} p(y=1|w,\check{c}) = \frac{\sum_{i=1}^{|W|} \#(\check{c}_i)^\alpha \cdot \sigma(V_w\widetilde{V}_{\check{c}_i})}{\sum_{i=1}^{|W|} \#(\check{c}_i)^\alpha} \tag{6.4}$$

Similar to PPMI (Eq. 2.3), our last proposed representation removes the negative values. The *Positive Reduced Explicit SkipGram (PRExpSG)* is defined as follows:

$$PRExpSG^c(w) = max(RExpSG^c(w), 0) \tag{6.5}$$

Setting the values to zero in PRExpSG facilitates the use of efficient data structures i.e. sparse vectors. We analyze the efficiency and effectiveness of the explicit representations in the next section.

6.1.2 Evaluation

To analyze the representations, we create a SkipGram model similar to the previous chapters with 300 dimensions on the Wikipedia dump file for August 2015 using the gensim toolkit [ŘS10]. As suggested by Levy et al. [LGD15], we use a window of 5 terms, negative sampling of $k = 10$, down sampling of $t = 10^{-5}$, a `cds` value of $\alpha = 0.75$, trained on 20 epochs, and filtering out terms with frequency less than 100. The final model contains 199851 terms. The same values are used for the common parameters in the PPMI and SPPMI representations.

We conduct our experiments on 6 term association benchmark collections. Each collection contains a set of term pairs where the association between each pair is assessed by several human annotators (*annotation score*). The evaluation is done by calculating the Spearman correlation between the list of pairs scored by similarity values versus by annotation scores. The collections used are: WordSim353 partitioned into Similarity and Relatedness [AAH+09]; MEN dataset [BTB14]; Rare Words dataset [LSM13]; SCWS [HSMN12]; and SimLex dataset [HRK15]. The statistics of the collections are shown in Table 6.1.

The evaluation results for the explicit representations as well as SG are reported in Table 6.2. The bold values show the best performing explicit representation and the values with underline refer to the best results among all representations. Based on the results, PRExpSG and SPPMI show very similar performance (in 3 benchmarks PRExpSG and in the other 3 SPPMI shows the best performance), both considerably outperforming

Table 6.1: Term association benchmarks.

Collection	# of Pairs
WordSim Similarity	203
WordSim Relatedness	252
MEN	3000
Rare	2034
SCWS	2003
SimLex	999

Table 6.2: Term association evaluation. Best performing among explicit/all embeddings are shown with bold/underline.

Method	Sparsity	WS Sim.	WS Rel.	MEN	Rare	SCWS	SimLex
PPMI	98.6%	.681	.603	.702	.309	.601	.284
SPPMI	99.6%	**.722**	**.661**	.704	.394	.571	**.296**
ExpSG	0%	.596	.404	.645	.378	.549	.231
RExpSG	0%	.527	.388	.606	.311	.507	.215
PRExpSG	94.1%	.697	.626	**.711**	**.406**	**.614**	.272
SG	0%	.770	.620	.750	.488	.648	.367

the other explicit representations. As also shown in previous studies [LGD15], SG in general performs better than the best performing explicit representations. The results confirm the quality of the PRExpSG model as a well-performing representation on term association benchmarks. Also looking at the sparsity ratio of the explicit representations, reported in Table 6.2, we observe that the PRExpSG and SPPMI representations are highly sparse, making them amenable to storage in volatile memory in practical scenarios.

In this section, we introduced the PRExpSG model and showed its strong performance in practice. In the next section, we use PRExpSG for gender bias quantification, and compare our results to the approach of Bolukbasi et al. [BCZ+16] conducted on SkipGram vectors. Using PRExpSG—an explicit representation variation of the SkipGram model—enables comparison between the two gender quantification approaches, since the PRExpSG representation exploited in our method is created from the SkipGram embedding, used in the approach of Bolukbasi et al..

6.2 Gender Bias Quantification with Explicit SkipGram

To study the gender bias in occupations, we prepare a list of 343 occupations, from which 26 are female-specific (e.g. 'congresswoman'), and 22 male-specific (e.g. 'congressman'), and the rest are gender neutral (e.g. 'nurse', 'dancer', 'bookkeeper'), listed in Table B.1, Table B.2, and Table B.3 respectively. In the following, we first explain in detail our

approach to gender bias quantification using the PRExpSG representation as well as the one used in Bolukbasi et al.. We then visualize the degrees of inclinations of the mentioned occupations to female and male by processing a corpus of Wikipedia.

6.2.1 Method

In Bolukbasi et al. [BCZ+16], the degree of gender bias of a word is measured using the following approach:

$$\widehat{\lambda}_f(w) = cosine(V_{she}, V_w), \quad \widehat{\lambda}_m(w) = cosine(V_{he}, V_w) \tag{6.6}$$

where $\widehat{\lambda}_f$ ($\widehat{\lambda}_m$) denotes the degree of bias of a word w (occupation in our case) to female (male), and V_{she} (V_{he}) is the vector representation of word 'she' ('he'), using the (dense) SkipGram model.

As mentioned in the introduction, due to the circularity in word embedding, using V_{she} and V_{he} does not provide a precise quantification of bias, as these vectors also contain concepts related to the occupations. To validate the existence of circularity, we calculate the value of $PRExpSG(she, c)$ and $PRExpSG(he, c)$ for each occupation (c indicating an occupation). Among the 343 occupations, we observe 123, and 168 values higher than zero for V_{she} and V_{he} respectively, indicating significant existence of occupation-related concepts in the gender vectors.

To address the issue raised by circularity, we use the set of gender-specific terms, provided by Bolukbasi et al. (refered to as *equalize pairs* in their work), to represent the gender-related concepts in language. We manually shortlist the terms by removing occupations (e.g. 'businessman' and 'businesswoman') and animals (e.g. 'colt' and 'filly'). The final list contains 32 female-specific terms (e.g. 'she', 'her', 'woman') and 32 equivalent male-specific terms (e.g. 'he', 'his', 'man'), denoted as S_f and S_m, shown in Table B.4, and Table B.5 respectively.

Using these lists of gender-concepts, we then create two new gender vectors V_{SHE} and V_{HE} in explicit space, defined as follows:

$$V^c_{SHE} = \begin{cases} PRExpSG^c(she) & c \in S_f \\ 0 & c \notin S_f \end{cases} \tag{6.7}$$

$$V^c_{HE} = \begin{cases} PRExpSG^c(he) & c \in S_m \\ 0 & c \notin S_m \end{cases} \tag{6.8}$$

where V^c denotes the value of the dimension (concept) c of the vector.

Given the new gender vectors, similar to Eq. 6.6 we define the new gender factors as follows:

$$\lambda_f(w) = cosine(V_{SHE}, V_w), \quad \lambda_m(w) = cosine(V_{HE}, V_w) \tag{6.9}$$

As the values of λ appear in a different range than the ones of $\widehat{\lambda}$, to make the approaches comparable, we apply Min-Max normalization on each approach, calculated over the gender factor values of all terms of the corpus.

Another important consideration in our analysis is to distinguish between truly gender-biased terms from low range values of gender factors (which can occur for every random term). To indicate the terms with no considerable inclinations to genders, we define gender-neutrality for a term when the difference between its gender factors is less than a threshold:

$$|\lambda_f - \lambda_m| < \zeta, \quad |\widehat{\lambda}_f - \widehat{\lambda}_m| < \widehat{\zeta} \tag{6.10}$$

To find such a threshold for each approach, since the number of gender-specific terms in English are limited, we assume that a randomly sampled term from the vocabulary is a gender-neutral term. This approach is similar to the one used in the Negative Sampling method. We can repeat this sampling for all the terms and calculate the expected values of ζ and $\widehat{\zeta}$ by averaging $|\lambda_f(w) - \lambda_m(w)|$ and $|\widehat{\lambda}_f(w) - \widehat{\lambda}_m(w)|$ respectively over the terms. In our experiments, this results in $\zeta = 0.046$ and $\widehat{\zeta} = 0.038$.

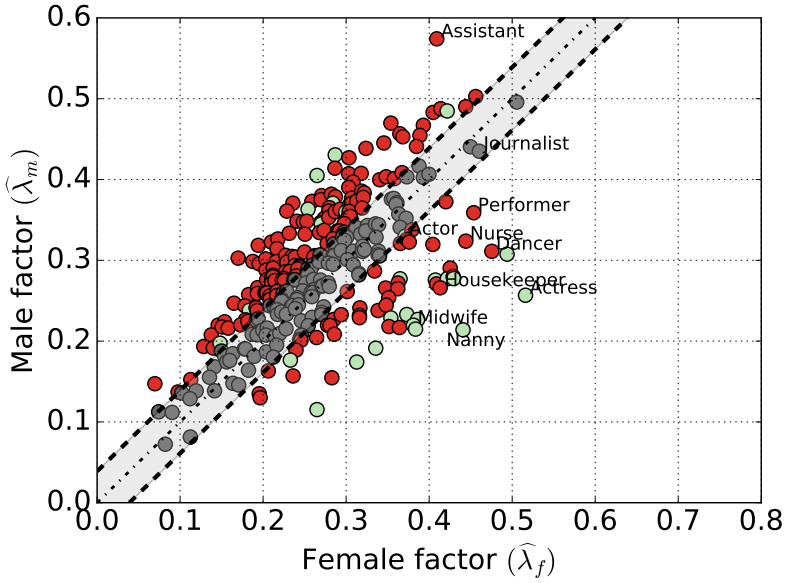
6.2.2 Quantification of Gender Bias in Wikipedia

The results of gender bias quantification methods, applied on the Wikipedia corpus are shown in Figure 6.1. Figures 6.1a and 6.1b depict the method used in Bolukbasi et al. and our approach to gender bias quantification method, respectively. In both figures, the gender-specific occupations are colored green, the gender-neutral ones red, and the gender-neutrality area gray.

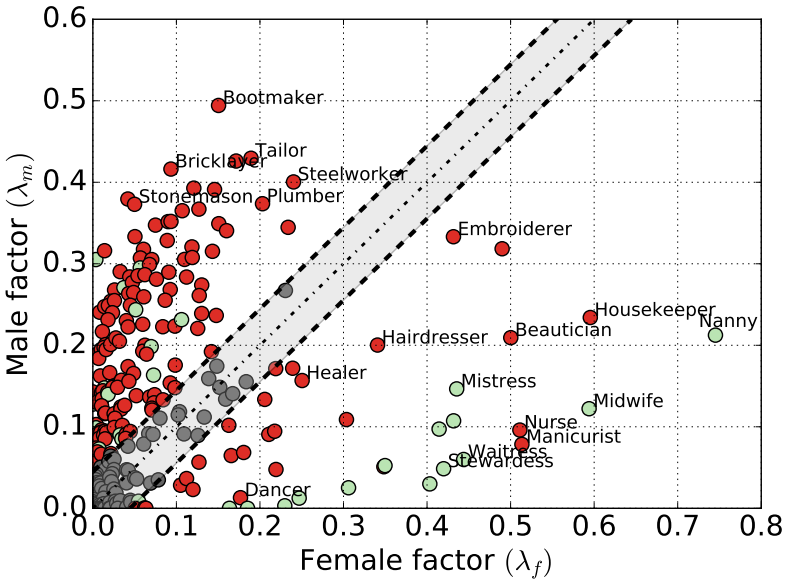
Comparing the two figures, we observe considerable differences between the gender bias, measured by the two approaches. To compare the approaches, we use WinoBias [ZWY+18], a recently introduced dataset which reports the degree of gender bias in 40 occupations, using the statistics gathered from the US Department of Labor. The degree of bias of each occupation to female in the dataset is the percent of people in the occupation who are reported as female (e.g. 90% of nurses are women). The dataset is shown in Table B.6.

We compare the results of the two approaches by calculating the correlation of female bias of these 40 occupations, quantified by each approach, with the values in the WinoBias dataset. The degree of bias to female for occupation w in our and Bolukbasi et al.'s approach is computed by $\lambda_f(w) - \lambda_m(w)$, and $\widehat{\lambda}_f(w) - \widehat{\lambda}_m(w)$, respectively. The evaluation makes the assumption that the bias in the real world is reflected in the text of Wikipedia.

The results of Spearman and Pearson correlations are shown in Table 6.3. For both Spearman and Pearson correlations, our approach shows higher correlation to the female bias values, provided by the WinoBias dataset. The results show that our approach more accurately resonates the state of gender bias in the real world, and is therefore a more precise method for bias quantification. In fact, our approach corrects the algorithmic bias in Bolukbasi et al.'s method, by addressing the issue of circular effect in word representations using explicit definition of gender-related concepts.



(a)



(b)

Figure 6.1: The inclination of occupations towards male and female genders. Gender-specific occupations are shown in green (light) and gender-neutral ones in red (dark). The gray area indicates gender-neutrality. (a) Method proposed in Bolukbasi et al. [BCZ+16] using dense SkipGram vectors. (b) Our approach to gender bias quantification using explicit SkipGram vectors.

Table 6.3: Spearman and Pearson Correlation results of female bias, quantified by our approach using PRExpSG representation and Bolukbasi et al.'s approach using SkipGram, to the female bias statistics of 40 occupations, provided by the WinoBias dataset [ZWY+18].

Method	Correlation	
	Spearman	Pearson
Bolukbasi et al. [BCZ+16]	0.38	0.39
Our Approach	**0.42**	**0.50**

Looking at the results of our approach in Figure 6.1b, it reveals an interesting pattern in gender bias for the gender-neutral occupations. The majority of these occupations are inclined towards the male factor while in general having weak bias. 'Bootmaker', 'tailor', and 'stonemason' are some of the male-biased occupations. On the other hand, there exist relatively few occupations with inclination to the female factor while some of them have very strong gender bias, for example gender-neutral occupations like 'housekeeper', 'nurse', and 'manicurist'. These observations provide a quantification of gender bias in machine learned representations and enable future automated gender debiasing.

6.3 Summary

In this chapter, we propose a method to create an explicit representation of the word2vec SkipGram model by capturing the probability of genuine co-occurrence of the terms. The proposed representation performs on par with the state of the art explicit representations on a set of term association benchmarks, and suggests a novel approach to interpret the vector embeddings of the SkipGram model.

We propose a method for quantifying gender bias using our explicit SkipGram representation, which addresses the problem of circular effect in word embeddings. Comparing our approach with the state of the art, we observe higher correlation between the values of female bias, quantified by our approach, and the actual statistics of gender bias of 40 occupations in the US labor market. Finally, looking at the gender bias results provided by our explicit SkipGram method, we observe a general tendency of the majority of jobs to the male factor while there is strong bias in a few specific occupations to the female factor. This study enables further research on algorithmic gender debiasing, especially by using explicit vectors.

CHAPTER 7

Cross-Lingual Word Sense Disambiguation with Word Embedding

In this chapter, we study an application of word embedding-based semantic approaches for Cross-Lingual Word Sense Disambiguation (CL-WSD). CL-WSD is the task of correctly translating an ambiguous term in a source language to a target language.

Typically, CL-WSD methods are classified into knowledge-based, supervised, and unsupervised. Knowledge-based approaches use available structured knowledge. Supervised approaches learn a computational model based on large amounts of annotated data. While these two approaches show excellent results in practice, they both have to face the knowledge acquisition bottleneck. This is a particular problem in specific domains or low-density languages. As an alternative, unsupervised approaches address CL-WSD using only information extracted from existing corpora, such as various term co-occurrence indicators.

For CL-WSD, two publicly available benchmarks, SemEval-2010 [LH10] and SemEval-2013 [LH13], provide an evaluation platform for word disambiguation from English to Dutch, German, Italian, Spanish, and French. We expand the SemEval-2013 test collection to the Persian (Farsi) language by creating a novel collection based on the CL-WSD SemEval format (explained in Appendix A).

Many participating systems in the SemEval tasks exploit parallel corpora, mainly Europarl [Koe05], to overcome the knowledge acquisition bottleneck [LHDC11, RLG13]. However, the approaches used in the tasks are not applicable for many languages and domains due to the scarcity of bilingual corpora. Persian, for instance, suffers from the lack of reliable and comprehensive knowledge resources as well as parallel corpora (Section A.1 reviews the available resources in Persian). In such cases, unsupervised methods

based on monolingual corpora (together with bilingual lexicon) are preferable, if not the only available option [SVT12]. For example, Bungum et al. [BGLM13] find the probable translations of a context in the source language and identify the best translation using a language model of the target language. Duque et al. [DAMR15] build a co-occurrence graph in the target language, and test a variety of graph-based algorithms for identifying the best translation match.

In terms of combining Word Sense Disambiguation (WSD) and word embedding, Chen et al. [CLS14] use knowledge-based WSD to identify distinct representations for different senses of the same term. Our approach for CL-WSD is the opposite of this: starting from word embedding representations, it identifies the similarity of the potential translations to the terms in their contexts and chooses the translation with the highest semantic similarity to its context.

In order to evaluate our approach, we use our new benchmark of English to Persian CL-WSD, and compare our approach and the CO-Graph system [DAMR15], observing the advantages of using word embedding in CL-WSD.

In terms of related work addressing the CL-WSD problem in Persian, Sarrafzadeh et al. [SYCA11] follows a knowledge-based approach by exploiting FarsNet [SHF+10]. However, since their evaluation collection is not available, the results are impossible to compare with other possible approaches.

The remainder of this chapter is organized as follows: Section 7.1 explains our unsupervised approach to English to Persian CL-WSD. We explain our experiment setup in Section 7.2, followed by discussing the results in Section 7.3. Finally, the study is concluded in Section 7.4.

7.1 Unsupervised CL-WSD Method

Our approach follows the main idea of the Lesk algorithm [Les86], namely that terms in a given context tend to share a common topic. We use word embedding to compute the semantic similarity between terms. We measure the similarity of each candidate translation of an ambiguous term to the translations of the context (the paragraph given by the task) and select the most similar translation to its context. Our CL-WSD approach is conceptually similar to the semantic matching algorithms, discussed in our previous studies [RBI+15, RBLH17, RBLH15].

To formulate our CL-WSD approach, let us define T as the list of translation sets for the terms in a context: $T = \{T_1, T_2, .., T_n\}$ where n is the number of terms in the context, and T_i is the set of possible translation terms for the i^{th} term in the context. For each translation term $t \in T_i$, we also have $P(t)$ as priori knowledge—an indicator of how frequent this particular translation is.

Given an embedding model in the target language, we compute the similarity of two translation terms t and $\bar{t}$ using their embedding vectors. However, in some cases the

translation t of one term in English may be two or more words in Persian (multi-word term), and since our word embedding model is generally created on the word level, we will have more than one vector. Therefore, assuming every term t as a set words w, we define a general similarity function between two translation terms as follows:

$$\text{sim}(t, \bar{t}) = \max_{w \in t, \bar{w} \in \bar{t}} (\cos(V_w, V_{\bar{w}})) \tag{7.1}$$

where V_w is the vector representation of the word, and cos is the cosine function.

Having a definition of similarity between two translation terms, we now move to defining the similarity between a candidate translation term of the ambiguous term and the list of translation sets T. We consider two ways to approach it:

The first, denoted as *RelAgg*, uses the *ContextVec* function to create a vector, representing the translated context terms in the target language. The ContextVec function is defined in Algorithm 7.1.

Algorithm 7.1: ContextVec

Input: translation term t, and the list of translation sets T
Output: vector representation of the context

$sumVec \leftarrow []$;
for $T_i \in T$ **do**
  $t^* \leftarrow \arg\max_{\bar{t} \in T_i} (\text{sim}(t, \bar{t}))$;
  $maxVec \leftarrow V_{t^*}$;
  $sumVec \leftarrow sumVec + maxVec$;
end
return norm($sumVec$);

The *norm* function in Algorithm 7.1 applies the Euclidean norm.

Given the vector representation of the context, RelAgg calculates the cosine between the vector of each candidate translation term t to the ContextVec(t, T), multiplied by the probability of the translation candidate $P(t)$, shown as follows:

$$\text{RelAgg}(t, T) = \cos(V_t, \text{ContextVec}(t, T))P(t) \tag{7.2}$$

The second approach, denoted as *RelGreedy*, searches among all the translation terms in all the sets T_i, and returns the value of the most similar translation term to the translation candidate. Similar to RelAgg, the final score is multiplied by the probability of the translation candidate. The RelGreedy approach is defined as follows:

$$\text{RelGreedy}(t, T) = \max_{T_i \in T} \left(\max_{\bar{t} \in T_i} (\text{sim}(t, \bar{t})) \right) P(t) \tag{7.3}$$

Finally, given the score of the similarity of each translation candidate t_i to its context using either RelAgg or RelGreedy, we can select the best translation among the candidates, as follows:

$$Result = \arg\max_{t_i} (\text{Rel}^*(t_i, T)) \tag{7.4}$$

where t_i is a translation candidate for the term with ambiguity, and Rel* is either RelAgg or RelGreedy.

7.2 Experiment Setup

Resources Similar to Jadidinejad et al. [JMD10], we use the PerStem tool [DL08] for stemming and TagPer [SMN12] for POS tagging of Persian language. We create a word2vec SkipGram model on a stemmed corpus of the Hamshahri collection [AAD$^+$09], containing 323616 documents of the Hamshahri newspaper (written in Persian). We use sub-sampling at $t = 10^{-4}$, the context windows of 5 terms, epochs of 25, term count threshold of 5.

Beside the monolingual word embedding, a bilingual lexicon is required for our unsupervised CL-WSD approach. While using parallel corpora is considered as a more effective method for creating lexica [DMRA15], due to the lack of reliable parallel corpora, we have to use a simple English to Persian dictionary. To have it in digital form, we use the online API of one of the Google Translate services[1]. The lexica also provides a translation probability rate, which we use as the $P(t)$ value[2].

Benchmark We use the novel English to Persian CL-WSD collection, described in Appendix A, which follows the format of SemEval-2013 test collection. The collection consists of 20 nouns, each with 50 cases (paragraphs) in English where the sense of each noun in its corresponding paragraphs is ambiguous. The aim of the benchmark is to find the correct Persian translations of the ambiguous terms.

Evaluation As the official evaluation measure of the SemEval 2013 CL-WSD task [LH13], we use the F score (harmonic mean of precision and recall), applied in two settings:

- *Best Result* (`Best`), in which a system suggests any number of translations for each target term, and the final score is divided by the number of these translations.
- *Out-Of-Five* (`OOF`) as a more relaxed evaluation setting, in which the system provides up to five different translations, and the best one among them is selected.

[1] Accessed on June 2015

[2] Available in `https://github.com/navid-rekabsaz/wsd_persian/tree/master/resources/dictionary`

Table 7.1: Results of F-measure on `OOF` and `Best` evaluation settings.

Setting	Method	F-measure
`OOF`	RelAgg	**0.502**
	RelGreedy	0.493
	CO-Graph Dijkstra [DAMR15]	0.441
	STD	0.418
`Best`	RelAgg	**0.188**
	RelGreedy	0.183
	CO-Graph Dijkstra [DAMR15]	0.174
	STD	0.158

Baselines The first—STD—is introduced in the SemEval 2013 CL-WSD task as a basic baseline. Similar to the original collection paper, to create the baseline we select the most common and the five most common translations for the `Best` and `OOF` settings respectively.

For the second baseline, we evaluate the Persian benchmark on the state-of-the-art unsupervised CL-WSD system, called CO-Graph [DAMR15]. The initial hypothesis for the CO-Graph system relies on the idea that words in a document tend to (statistically) adopt a related sense. The system first creates a graph of connections between the words, using the documents in the collection, and then applies different algorithms (Dijkstra, Community-based, Static PageRank, Personalized PageRank) to disambiguate the words based on their contexts. The construction of the graph is based on the statistical significance (p-value) of the co-occurrences of the words in the same documents.

The CO-Graph system offers competitive results in the SemEval 2013 CL-WSD tasks, for all the proposed languages. It outperforms all of the unsupervised participating systems using only monolingual corpora, and even most of the ones which use parallel corpora or knowledge resources. As our English-Farsi test collection is also created based on the SemEval 2013 task, we find this system as a strong baseline for our experiments. To evaluate the CO-Graph system on the Persian benchmark, we first create the graph using the articles of the Hamshahri collection, each as a document. In the construction of the graph, we only take into account the nouns by POS tagging. After evaluating various algorithms, we find the Dijkstra algorithm together with p-value=10^{-6} as the best performing approach.

Preprocessing We apply POS tagging on the English sentences of the SemEval 2013 CL-WSD task and only select the verbs and nouns as the context of the ambiguous terms. We then lemmatize the context terms using WordNetLemmatizer of the NLTK toolkit.

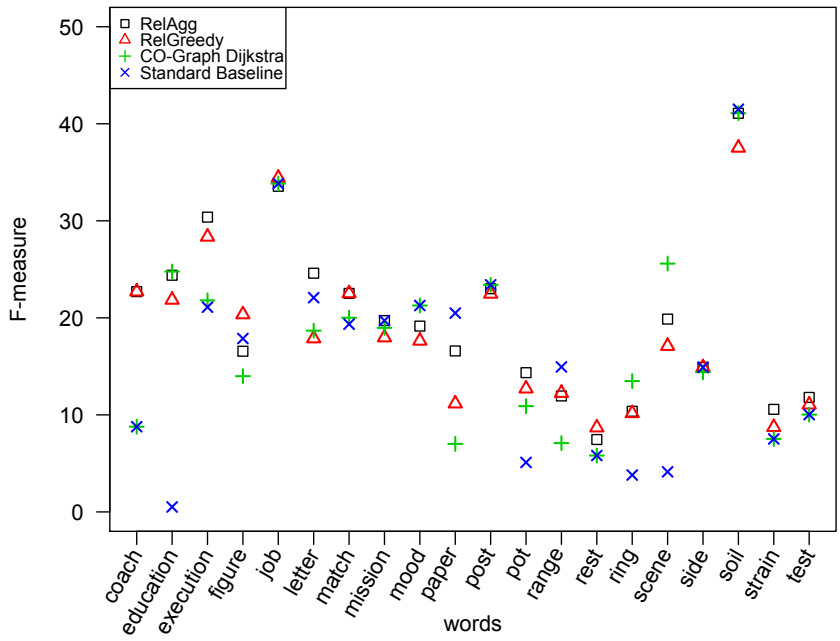


Figure 7.1: F-measure results (multiplied by 100) of the `Best` evaluation setting for 20 ambigous terms of the SemEval 2013 CL-WSD task

7.3 Results and Discussion

Table 7.1 shows the F-measure results of RelAgg and RelGreedy as well as the baselines on the `OOF` and `Best` evaluation settings. The results for both evaluation settings show that our approach outperforms the standard and the CO-Graph baselines. Comparing the approaches, we observe similar results for the RelAgg and RelGreedy methods, while RelAgg has slightly better performance, specially in the `OOF` evaluation setting.

In Figure 7.1, we compare the effectiveness of our methods for each ambiguous term of the task with the baselines in the `Best` setting as the more challenging one. The results show that while for most terms, our approach outperforms the standard baseline as well as the CO-Graph system, none of the systems can outperform the standard baseline for the terms 'mood' and 'side'. Analyzing the results of these terms, we observe that in some sentences, none of the nouns and verbs in the context share any common topic with senses of the ambiguous terms. For example, using only the semantics of the nouns and verbs in the context, the correct sense of 'mood' cannot be distinguished in either of the sentences: 'it reflected the *mood* of the moment' (state of the feeling) and 'a general *mood* in Whitehall' (inclination, tendency) . Similar cases are observed for the term 'side': e.g., 'both *sides* reaffirmed their commitment' (groups opposing each other) in comparison

to 'at the *side* of the cottage' (a position to the left or right of a place). While these examples show the limitations of the context-based methods, the overall results show the ability of word embedding and statistical-based approaches for the CL-WSD tasks, specially in the absence of reliable resources.

7.4 Summary

In this chapter, we study the application of word embedding-based methods in unsupervised Cross Language Word Sense Disambiguation (CL-WSD) when translating an English noun, appeared in a paragraph, to Persian. Our semantic approach uses embedding of the candidate translations as well as translated context terms to calculate the semantic similarity of each translation to its context. The proposed approach outperforms both the CO-Graph system—a state-of-the-art system in unsupervised CL-WSD—as well as the standard baseline.

We however observe fundamental limitations of the methods based exclusively on context as bag of words when none of the context terms share any semantic topic with the ambiguous terms. Despite this fact, the current work offers a possible solution for all languages/domains with scarce knowledge-based or parallel corpora resources, by exploiting the use of a monolingual corpus together with a simple bilingual lexicon.

CHAPTER 8

Sentiment Analysis with Generalized Translation Models

In this chapter, we investigate the application of our word embedding-based methods in another domain, namely in the financial sentiment analysis for volatility prediction. Financial volatility is an essential indicator of instability and risk of a company, sector or economy. Volatility forecasting has gained considerable attention during the last three decades. In addition to using historic stock prices, recent approaches to volatility prediction use sentiment analysis to exploit various text resources, such as financial reports [KLR+09,WTLC13,TW14,NH15], news [KZP14,DZLD15], message boards [NS15], and earning calls [WH14].

An interesting resource of textual information are U.S. companies' annual disclosures, known as *10-K filing* reports. They contain comprehensive information about the companies' business as well as risk factors. Specifically, section *Item 1A - Risk Factors* of the reports contains information about the most significant risks for the company. These reports are however long, redundant, and written in a style that makes them complex to process. Dyer er al. [DLSL16] notes that: *"10-K reports are getting more redundant and complex* [...] *(it) requires a reader to have 21.6 years of formal education to fully comprehend"*. Dyer et al. also analyse the topics discussed in the reports and observe a constant increase over the years in both the length of the documents as well as the number of topics. They claim that the increase in length is not the result of economic factors but is due to verboseness and redundancy in the reports. They suggest that only the risk factors topic appears to be useful and informative to investors. Their analysis motivates us to study the effectiveness of the Risk Factors section for volatility prediction.

The research in this chapter builds on previous studies on volatility prediction and information analysis of 10-K reports using sentiment analysis [KLR+09, TW14, WTLC13, NH15, Li10, CCD+14], in the sense that since the reports are long (average length of 5000

terms), different approaches are required, compared with studies of sentiment analysis on short-texts. Such previous studies on 10-K reports have mostly used the data before 2008 and there is little work on the analysis of the informativeness and effectiveness of the recent reports with regards to volatility prediction. We will indeed show that the content of the reports changes significantly not only before and after 2008, but rather in a cycle of 3-4 years.

In terms of use of the textual content for volatility prediction, the work in this chapter shows that the Generalized Translation model as a term weighting scheme has a significantly positive impact on prediction accuracy. The most recent study on the topic [TW14] used related terms obtained by word embeddings to expand the lexicon of sentiment terms. In contrast, as described in Chapter 3, we define the weight of each lexicon term by extending it to the similar terms in the document. The significant improvement of this approach for document retrieval by capturing the information of similar terms motivates us to apply it on sentiment analysis. We extensively evaluate various state-of-the-art sentiment analysis methods to investigate the effectiveness of our approach.

In addition to text, factual market data (i.e. historical prices) provide valuable resources for volatility prediction e.g. in the framework of GARCH models [Eng82]. An emerging question is how to approach the combination of the textual and factual market information. We propose various methods for this issue and show the performance and characteristics of each.

The financial system covers a wide variety of industries, from daily-consumption products to space mission technologies. It is intuitive to consider that the factors of instability and uncertainty are different between the various sectors while similar inside them. We therefore also analyze the sentiment of the reports of each sector separately and study their particular characteristics.

The present study shows the value of information in the 10-K reports for volatility prediction. Our proposed approach to sentiment analysis significantly outperforms state-of-the-art methods [KLR$^+$09, TW14, WTLC13]. We also show that performance can be further improved by effectively combining textual and factual market information. In addition, we shed light on the effects of tailoring the analysis to each sector: despite the reasonable expectation that domain-specific training would lead to improvements, we show that our sector-agnostic model generalizes well and outperforms sector-specific trained models.

The remainder of the chapter is organized as follows: in the next section, we review the state-of-the-art and related studies to sentiment-based volatility prediction. Section 8.2 formulates the problem, followed by a detailed explanation of our approach in Section 8.3. We explain the dataset and settings of the experiments in Section 8.4, followed by the full description of the experiments in Section 8.5. We conclude the chapter in Section 8.6.

8.1 Related Work to Volatility Prediction

Market prediction has been attracting much attention in recent years. Kazemian et al. [KZP14] use sentiment analysis for predicting stock price movements in a simulated security trading system using news data, showing the advantages of the method against simple trading strategies. Ding et al. [DZLD15] address a similar objective while using deep learning to extract and learn events in the news. Xie et al. [XPW13] introduce a semantic tree-based model to represent news data for predicting stock price movement. Luss et al. [Ld15] also exploit news in combination with return prices to predict intra-day price movements. They use the Multi Kernel Learning (MKL) algorithm for combining the two features. The combination shows improvement in final prediction in comparison to using each of the features alone. Motivated by this study, we investigate the performance of the MKL algorithm as one of the methods to combine the textual with non-textual information. Other data resources, such as stocks' message boards, are used by Nguyen and Shirai [NS15] to study topic modeling for aspect-based sentiment analysis. Wang and Hua [WH14] investigate the sentiment of the transcript of earning calls for volatility prediction using the Gaussian Copula regression model.

While the mentioned studies use short-length texts (sentence or paragraph level), approaching long texts (document level) for market prediction is mainly based on n-gram bag of words methods. Nopp and Hanbury [NH15] study the sentiment of banks' annual reports to assess banking systems risk factors using a finance-specific lexicon, provided by Loughran and McDonald [LM11], in both unsupervised and supervised manner.

More directly related to the informativeness of the 10-K reports for volatility prediction, Kogan et al. [KLR$^+$09] use a linear Support Vector Machine (SVM) algorithm on the reports published between 1996–2006. Wang et al. [WTLC13] improve upon this by using the Loughran and McDonald lexicon, observing improvement in the prediction. Later, Tsai and Wang [TW14] apply the same method as Wang et al. [WTLC13] while additionally using word embedding to expand the financial lexicon. We reproduce all the methods in these studies, and show the advantage of our sentiment analysis approach.

8.2 Problem Formulation

In this section, we formulate the volatility forecasting problem and the prediction objectives of our experiments. Similar to previous studies [CSS12, KLR$^+$09, TW14], volatility is defined as the natural log of the standard deviation of (adjusted) return prices in a window of τ days. This definition is referred to as standard volatility [LH11] or realized volatility [LT13], defined as follows:

$$v_{[s,s+\tau]} = ln\left(\sqrt{\frac{\sum_{t=s}^{s+\tau}(r_t - \bar{r})^2}{\tau}}\right) \tag{8.1}$$

where r_t is the return price and $\bar{r}$ the mean of return prices. The return price is calculated by $r_t = ln(P_t) - ln(P_{t-1})$, where P_t is the (adjusted) closing price of a given stock at the

trading date t.

Given an arbitrary report i, we define a prediction label y_i^k as the volatility of the stock of the reporting company in the kth quarter-sized window starting from the issue date of the report s_i:

$$y_i^k = v_{[s_i+64(k-1),s_i+64k]} \tag{8.2}$$

Every quarter is considered as per convention, 64 working days, while the full year is assumed to have 256 working days.

We use 8 learners for labels y^1 to y^8. For brevity, unless otherwise mentioned, we report the volatility of the first year by calculating the mean of the first four quartiles after the publication of each report.

8.3 Methodology

We first describe our text sentiment analysis methods, followed by the features obtained from factual market data, and finally explain the methods to combine textual and market feature sets.

8.3.1 Sentiment Analysis

Similar to previous studies [NH15, WTLC13], we extract the keyword set from a finance-specific lexicon [LM11] using the positive, negative, and uncertain groups, stemmed using the Porter stemmer. We refer to this keyword set as `Lex`. Tsai et al. [TW14] expanded this set by adding the top 20 related terms to each term to the original set. The related terms are obtained using the word2vec model, built on the corpus of all the reports, with cosine similarity. We also use this expanded set in our experiments and refer to it as `LexExt`.

The following term weighting schemes are commonly used in IR and we consider them as well in our study:

$TC:$ $\quad log(1 + tf_d(t))$

$TF:$ $\quad \frac{log(1+tf_d(t))}{\|d\|}$

$TFIDF:$ $\quad \frac{log(1+tf_d(t))}{\|d\|} log\left(1 + \frac{|d|}{df(t)}\right)$

$BM25:$ $\frac{(k+1)\overline{tf_d(t)}}{k+\overline{tf_d(t)}}$, $\quad \overline{tf_d(t)} = \frac{tf_d(t)}{(1-b)+b\frac{|d|}{avgdl}}$

where as before, $tf_d(t)$ is the number of occurrences of keyword t in report d, $\|d\|$ denotes the Euclidean norm of the keyword weights of the report, $|d|$ is the length of the report (number of the terms in the report), $avgdl$ is the average document length, and finally k and b are parameters. For them, we use the settings used in Chapter 3, i.e. $k = 1.2$ and $b = 0.65$.

In addition to the standard weighting schemes, we use the Generalized Translation models (presented in Chapter 3). We define the extended versions of the standard weighting schemes as $\widehat{TC}$, $\widehat{TF}$, $\widehat{TFIDF}$, and $\widehat{BM25}$ by replacing $tf_d(t)$ with $\widehat{tf_d}(t)$ (Eq. 3.3) in each of the schemes. As in the previous chapters, we select the list of similar terms to the keyword t from a word embedding model, using cosine as similarity measure and threshold of 0.70.

We also test the effectiveness of the Extended Translation models as weighting schemes. However, in practice we observe very similar results to the Generalized Translation models, and due to the smaller complexity of the latter, we only report the results of the weighting schemes based on the Generalized Translation models.

The feature vector generated by the weights of the `Lex` or `LexExt` lexicons is highly sparse, as the number of dimensions is larger than the number of data-points. We therefore reduce the dimensions to 400 by applying Principle Component Analysis (PCA). The dimension size 400 shows the best result from a range of dimensions from 50 to 1000 when evaluating on a randomly selected validation set with 20% of the size of the training data.

Given the final feature vector x with l dimensions, we apply SVM as a well-known method for training both regression and classification methods. Similar to previous studies [TW14, KLR+09], we set the parameters of the SVM to $C = 1.0$ and $\epsilon = 0.1$. We evaluate the performance of various kernels on the mentioned validation set, observing better performance of the Radial Basis Function (RBF) kernel in comparison to linear and cosine kernels and is therefore used in this work.

In addition, motivated by Moraes et al. [MVN13], we tested the effectiveness of neural network methods for volatility prediction. We tried neural network architectures with one or two hidden layers, each layer with either 400 or 500 nodes. For regularization, we tried the early-stopping, regularization term, and dropout methods. All the networks use *tanh* for activation function, and learning rate of 0.001 in gradient decent. However, none of the mentioned variations of the neural networks models could provide better results than the SVM regressors. Therefore, for this work, we only report the SVM methods.

8.3.2 Market Features

In addition to textual features, we define three features using the factual market data and historical prices—referred to as *market features*—as follows:

Current Volatility is calculated on the window of one quartile before the issue date of the report: $v_{[s_i-64,s_i]}$.

GARCH [Bol86] is a common econometric time-series model used for predicting stock price volatility. We use a GARCH (1, 1) model, trained separately for each report on intra-day return prices. We use all price data available before the issue date of the report for fitting the model. GARCH (1, 1) predicts the volatility of the next day by looking at the previous day's volatility. When forecasting further than one day into the future

Table 8.1: The financial sectors of companies and their abbreviations.

Energy	ene	Basic Industries	ind	Finance	fin
Technology	tech	Miscellaneous	misc	Consumer Non-Durables	n-dur
Consumer Durables	dur	Capital Goods	capt	Consumer Services	serv
Public Utilities	pub	Health Care	hlth		

one needs to use the model's own predictions in order to be able to make predictions for more than one day ahead. When forecasting further into the future these conditional forecasts of the variance will converge to a value called *unconditional variance*. As our forecast period is one quarter, we will approximate the volatility of future quarters with the unconditional variance.

Sector is the sector that the corresponding company of the report belongs to. The sectors and their abbreviations used in this paper, are listed in Table 8.1[1]. The feature is converted to numerical representation using one-hot encoding.

8.3.3 Feature Fusion

To combine the text and market feature sets, the first approach, used also in previous studies [KLR+09, WTLC13] is simply joining all the features in one feature space. In the context of multi-model learning, the method is referred to as *early fusion*.

In contrast, *late fusion* approaches first learn a model on each feature set and then use/learn a meta model to combine their results. As our second approach, we use *stacking* [Wol92], a special case of late fusion. In stacking, we first split the training set into two parts (70%-30% portions). Using the first portion, we train separate machine learning models for each of the text and market feature sets. Next, we predict labels of the second portion with the trained models and finally train another model to capture the combinations between the outputs of the base models. In our experiments, the final model is also trained with SVM with RBF kernel.

Stacking is computationally inexpensive. However, due to the split of the training set, the base models or the meta model may suffer from lack of training data. A potential approach to learn both the feature sets in one model is the Multi Kernel Learning (MKL) method.

The MKL algorithm (also called *intermediate fusion* [Nob04]) extends the kernel of the SVM model by learning (simultaneous to the parameter learning) an optimum combination of several kernels. Lanckriet et al. [LCB+04] formulates the MKL algorithm as follows:

$$K^* = \sum_i d_i K_i \quad \text{where } \sum_i d_i = 1,\ d_i \geq 0 \tag{8.3}$$

[1] We follow the NASDAQ categorization of sectors.

Table 8.2: Number of reports in the dataset per year.

Year	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	Sum
# of Reports	646	664	697	800	863	927	887	959	1051	1090	8584

where K_i is a predefined kernel. Gönen and Alpaydın [GA11] mention two uses of MKL: learning the optimum kernel in SVM, and combining multiple modalities (feature sets) via each kernel. Our objective in this work is the latter, namely combining text and market modalities (two kernels). To keep it consistent with other SVM-based approaches, we use the RBF kernel function for both the text and market feature sets.

The optimization of the MKL approach however can be computationally challenging. We use the *mklaren* method [SC16] which has linear complexity in the number of data instances and kernels, and has shown better performance in comparison with the recent multi kernel approximation approaches.

8.4 Experiment Setup

In this section, we first describe the data, followed by introducing the baselines. We report the parameters applied in various algorithms and describe the evaluation metrics.

Dataset We download the reports of companies of the U.S. stock markets from 2006 to 2015 from the U.S. Securities and Exchange Commission (SEC) website[2]. We remove HTML tags and extract the text parts. We extract the Risk Factors section using term matching heuristics. Finally, the texts are stemmed using the Porter stemmer. The statistics of the collection per year is shown in Table 8.2.

We calculate the volatility values (Eq 8.1) and the volatility of the GARCH model based on the stock prices, collected from the Yahoo website. Similar to Kogan et. al [KLR+09], we assume the volatility values greater/smaller than the mean plus/minus three times the standard deviation of all the volatility values as outliers and filter them out[3].

Baselines **GARCH:** as the GARCH model only uses historical prices of the stocks for prediction, we use it as a baseline to compare the effectiveness of text-based methods with mainstream approaches.

Market: uses all the market features, listed in Section 8.3.2. We train a SVM model with RBF kernel on these features to predict volatility.

[2] `https://www.sec.gov`

[3] The complete dataset is available in `http://ifs.tuwien.ac.at/~admire/financialvolatility`

Wang et al. [WTLC13]: they use the `Lex` keyword set with TC weighting scheme and the SVM method. They combine the textual features with current volatility using the early fusion method.

Tsai et al. [TW14]: similar to Wang et al. [WTLC13], while they use the `LexExt` keyword set.

Evaluation Metrics As a common metric in volatility prediction, we use the r^2 metric (square of the correlation coefficient) for evaluation:

$$r^2 = \left(\frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \right)^2 \tag{8.4}$$

where $\hat{y}_i$ is the predicted value, y_i denotes the labels and $\bar{y}$, their mean. The r^2 metric is between 1 and 0, indicating the proportion of variance in the labels explained by the prediction. An alternative metric, used in previous studies [WTLC13, TW14, KLR+09] is Mean Squared Error $MSE = \sum_i (\hat{y}_i - y_i)^2/n$. However, especially when comparing models, applied on different test sets (e.g. performance of first quartile with second quartile), r^2 has better interpretability since it is independent of the scale of y. We use r^2 in all the experiments while the MSE measure is reported only when the models are evaluated on the same test set.

8.5 Experiments and Results

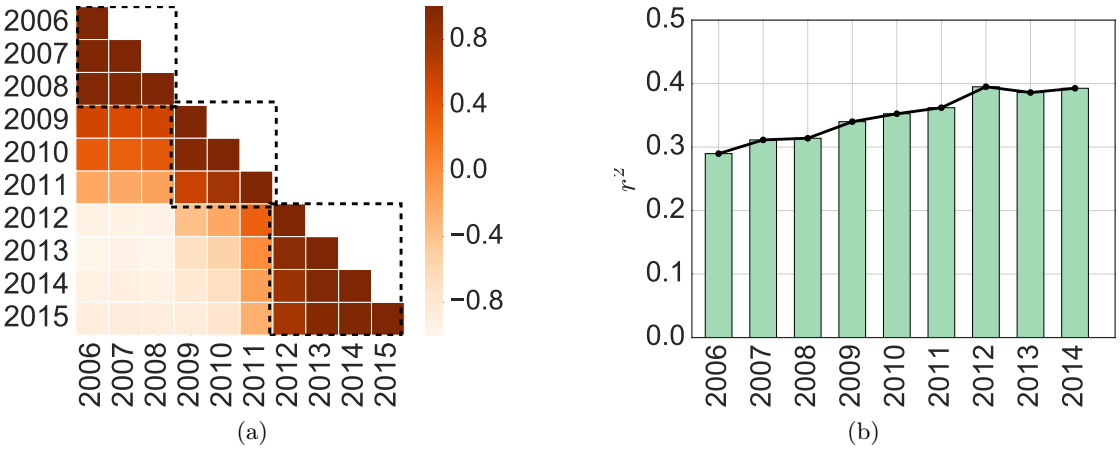
In this section, first we analyse the contents of the reports, followed by studying our sentiment analysis methods for volatility prediction. Finally, we investigate the effect of sentiment analysis of the reports in different industry sectors.

8.5.1 Content Analysis of 10-K Reports

Let us start our experiment with observing changes in the feature vectors of the reports over the years. To compare them, we use the state-of-the-art sentiment analysis method, introduced by Tsai and Wang [TW14]. We first represent the feature vector of each year by calculating the centroid (element-wise mean) of the feature vectors of all reports published that year and then calculate the cosine similarity of each pair of centroid vectors, for the years 2006–2015.

Figure 8.1a shows the similarity heat-map for each pair of the years. We observe a high similarity between three ranges of years: 2006–2008, 2009–2011, and 2012–2015. These considerable differences between the centroid reports in years across these three groups hints at probable issues when using the data of the older years as training data for predicting the volatility of more recent years.

To validate this, we apply 5-fold cross validation (folds are formed randomly), first on all the data (2006–2015), and then on smaller sets by dropping the earliest year i.e. the



(a) (b)

Figure 8.1: (a) Cosine similarity between the centroid vectors of the years. (b) Volatility prediction performance when using reports from the specified year to 2015

next subsets use the reports 2007–2015, 2008–2015 and so forth. The results of the r^2 measure are shown in Figure 8.1b. We observe that by dropping the oldest years one by one (from left to right in the figure), the performance starts improving. We argue that this improvement is due to the reduction of noise in data, noise caused by conceptual drifts in the reports as also mentioned by Dyer et al. [DLSL16]. In fact, although in machine learning in general using more data results in better generalization of the model and therefore better prediction, the reports of the older years introduce noise.

As shown, the most coherent and largest data consists of the subset of the reports published between 2012 to 2015. This subset is also the most recent cluster and presumably more similar to the future reports. Therefore, in the following, we only use this subset, which consists of 3892 reports, belonging to 1323 companies.

8.5.2 Volatility Prediction

Given the dataset of the 2012–2015 reports, we try all combinations of different term weighting schemes using the `LexExt` keyword set. All weighting schemes are then combined with the market features with the introduced fusion methods. The prediction is done with 5-fold cross validation. The averages of the results of the first four quartiles (first year) are reported in Table 8.3. To make showing the results tractable, we use the best fusion (stacking) for the weighting schemes and the best scheme ($\widehat{BM25}$) for fusions.

Regarding the weighting schemes, $\widehat{BM25}$, $BM25$, and $\widehat{TC}$ show the best results. In general, the extended schemes (with hat) improve upon their normal forms. For the feature fusion methods, stacking outperforms the other approaches in both evaluation measures. MKL has better performance than early fusion on r^2 and close results with MSE, while it has the highest computational complexity among the methods.

Table 8.3: Performance of sentiment analysis methods for the first year.

Component	Method	Text		Text+Market	
		(r^2)	(MSE)	(r^2)	(MSE)
Weighting Schema (+Stacking)	$\widehat{\boldsymbol{BM25}}$	**0.439**	**0.132**	**0.527**	**0.111**
	$BM25$	0.433	0.136	0.523	0.114
	$\widehat{TC}$	0.427	0.136	0.517	0.115
	TC	0.425	0.137	0.521	0.114
	$\widehat{TFIDF}$	0.301	0.166	0.502	0.118
	$TFIDF$	0.264	0.189	0.497	0.119
	$\widehat{TF}$	0.218	0.190	0.495	0.120
	TF	0.233	0.200	0.495	0.120
Feature Fusion ($+\widehat{BM25}$)	**Stacking**	-	-	**0.527**	**0.111**
	MKL	-	-	0.488	0.126
	Early Fusion	-	-	0.473	0.125

Table 8.4: Performance of the methods using 5-fold cross validation.

Method		(r^2)	(MSE)
	GARCH	0.280	0.170
Text	Wang [WTLC13]	0.345	0.154
	Tsai [TW14]	0.395	0.142
	Our method	**0.439**	**0.132**
	Market	0.485	0.122
Text+Market	Wang [WTLC13]	0.499	0.118
	Tsai [TW14]	0.484	0.122
	Our method	**0.527**	**0.111**

Based on these results, as our best performing approach in the remainder of the chapter, we use $\widehat{BM25}$ (with `LexExt` set), reduced to 400 dimensions and stacking as the fusion method. Table 8.4 summarizes the results of our best performing method compared with previously existing methods. Our method outperforms all state-of-the-art methods both when using textual features only as well as a combination of textual and market features.

Let us now take a closer look at the changes in the performance of the prediction in time. The results of 5-fold cross validation on the dataset of the reports, published between 2012–2015 are shown in Figure 8.2a. The X-axis shows eight quartiles after the publication date of the report. For comparison, the GARCH and only market features are depicted with dashed lines.

As shown, the performance of both GARCH and Market methods (approaches without text features) decrease faster in the later quartiles since the historical prices used for

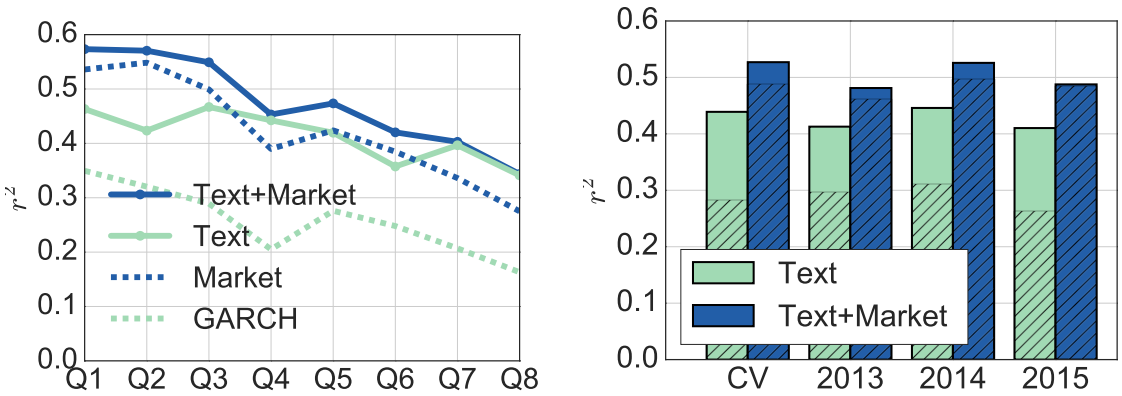


Figure 8.2: (a) Performance of our approach on 8 quartiles using the Text and Text+Market feature sets. The dashed lines show the market-based baselines. (b) Performance of volatility prediction of each year given the past data. CV indicates the cross validation scenario. The hashed areas show corresponding methods without text data (GARCH for Text, Market for Text+Market).

prediction become less relevant as time goes by. Using only text features (Text), we see a roughly similar performance between the first four quartiles (first year), while the performance, in general, slightly decreases in the second year. By combining the textual and market features (Text+Market), we see a consistent improvement in comparison to each of them alone. In comparison to using only market features, the combination of the features shows more stable results in the later quartiles. These results support the informativeness of the 10-K reports to more effectively foresee volatility in long-term windows.

While the above experiments are based on cross-validation, for the sake of completeness it is noteworthy to consider the scenarios of real-world applications where the future prediction is based on past data. We therefore design three experiments by considering the reports published in 2013, 2014, and 2015 as test set and the reports published before each year as training set (only 2012, 2012–2013, and 2012–2014 respectively). The results of predicting the reports of each year together with the cross validation scenario (CV) are shown in Figure 8.2b. The hashed areas indicate the corresponding methods without text features, namely GARCH and Market for the Text and Text+Market feature sets, respectively. While the performance becomes slightly worse in the target years 2013 and 2015, in general the combination of textual and market features can explain approximately half of volatility in the financial system.

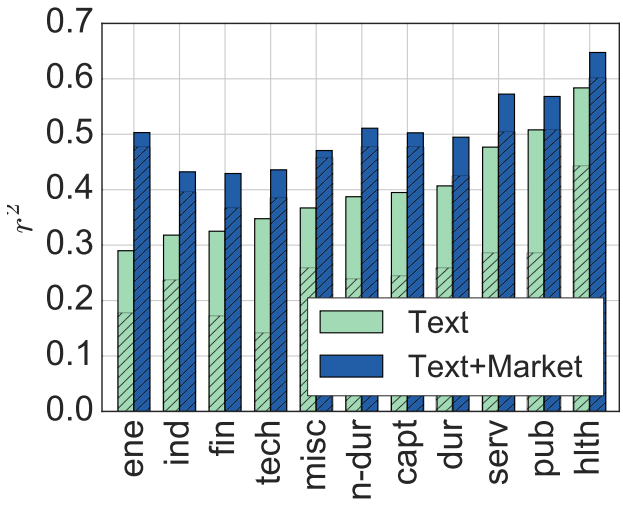


Figure 8.3: Performance per sector. Abbreviations are defined in Section 8.3.2

Table 8.5: Number of reports per sectors

ene	ind	hlth	fin	tech	pub	n-dur	dur	capt	serv	misc
187	160	305	847	408	217	151	115	255	639	153

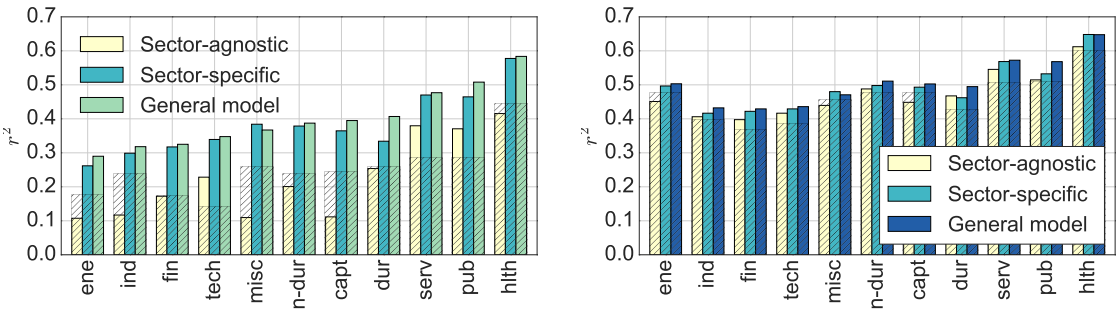
8.5.3 Sectors

Corporations in the same sector share not only similar products or services but also risks and instability factors. Considering the sentiment of the financial system as a homogeneous body may neglect the specific factors of each sector. We therefore set out to investigate the existence and nature of these differences.

We start by observing the prediction performance on different sectors: We use our method from the previous section, but split the test set across sectors and plot the results in Figure 8.3. As before, the hashed areas indicate the GARCH and Market methods for the Text and Text+Market feature sets, respectively. We observe considerable differences between the performance of the sectors, especially when using only sentiment analysis methods (i.e. only text features).

Given these differences and also the probable similarities between the risk factors of the reports in the same sector, a question immediately arises: can training different models for different sectors improve the performance of prediction?

To answer it, for each sector, we train a model using only the subset of the reports in that sector and use 5-fold validation to observe performance. We refer to these models as sector-specific in contrast to the general model, trained on all the data. Figures 8.4a and 8.4b compare their results: we can see that the sector-specific bars are lower than the general model ones. This is to some extent surprising, as one would expect that



(a) Text (b) Text+Market

Figure 8.4: Results when retraining on sector-specific subsets versus the general model and versus subsets of the same size but sector-agnostic. The hashed area in (a) indicates the GARCH and in (b) the Market baseline.

domain-specific training would improve the performance of sentiment analysis in text. However, we need to consider the size of the training set. By training on each sector we have reduced the size of our training sets to those reported in Table 8.5. To verify the effect of the size of training data, we train a sector-agnostic model for each sector. Each sector-agnostic model is trained by random sampling of a training set of the same size as the set available for its sector from all the reports, but evaluated–similar to sector-specific models–on the test set of the sector. Figures 8.4a and 8.4b also plot the results of the sector-agnostic models.

The large performance differences between sector-agnostic and -specific show the importance of sector-specific risk factors. Since the data for training in each sector is too small, we expect that as additional data is accumulated, we can further improve on the results by training on different sectors independently.

We continue by examining some examples of essential terms in sectors. To address this, we have to train a linear regression method on all the reports of each sector, without using any dimensionality reduction. Linear regression without dimensionality reduction has the benefit of interpretability: the coefficient of each feature (i.e. term in the lexicon) can be seen as its importance with regards to volatility prediction. After training, we observe that some keywords e.g. ‘crisis’, or ‘delist’ constantly have high coefficient values in the sector-specific as well as general model. However, some keywords are particularly weighted high in specific-sector models.

For instance, the keyword ‘fire’ has a high coefficient in the energy sector, but very low in the others. The reason is due to the problem of ambiguity i.e. in the energy sector, ‘fire’ is widely used to refer to ‘explosion’ e.g. ‘fire and explosion hazards’ while in the lexicon, it is stemmed from ‘firing’ and ‘fired’: the act of dismissing from a job. This latter sense of term is however weighted as a low risk-sensitive keyword in the other sectors. Such an ambiguity can indeed be mitigated by sector-specific models since the variety of the terms’ senses are more restricted inside each sector. Another example is an interesting

observation on the term 'beneficial'. The term is introduced as a positive sentiment in the lexicon while it gains highly negative sentiments in some sectors (health care and basic industries). Investigating in the reports, we observe the broad use of the expression 'beneficial owner' which is normally followed by risk-full sentences since the beneficial owners can potentially influence shareholders' decision power.

8.6 Summary

In this chapter, we study the sentiment of recent 10-K annual disclosures of companies in stock markets for forecasting volatility. Our bag-of-words sentiment analysis approach benefits from the Generalized Translation models which use word embeddings to extend the weight of the terms to the similar terms in the document. Additionally, we explore fusion methods to combine the text features with factual market features, achieved from historical prices i.e. GARCH prediction model, and current volatility. In both cases, our approach outperforms state-of-the-art volatility prediction methods with 10-K reports and demonstrates the effectiveness of sentiment analysis in long-term volatility forecasting.

In addition, we study the characteristics of the companies' reports in each financial sector with regard to risk-sensitive terms. Our analysis shows that reports in same sectors considerably share particular risk and instability factors. However, despite expectations, training different models on different sectors does not improve performance compared to the general model. We trace this to the size of the available data in each sector, and show that there are still benefits in considering sectors, which could be further explored in the future.

CHAPTER 9

Conclusion

Statistical word representation models are an essential pillar of text and language processing, and have been the focus of research for decades. Motivated by recent advancements in neural representation models, in this thesis, we investigate novel methodologies to exploit word representation models in various text analysis tasks such as Information Retrieval (IR), sentiment analysis, gender bias detection, and Cross-Lingual Word Sense Disambiguation (CL-WSD). Our basic and applied research on word representation models provides remarkable insights on the statistical semantics approaches in text and language processing.

The first part of the thesis carries out basic research on the integration of word embedding in IR models, the selection of related terms through exploration of the embedding space, and the fusion of similarities from window- and document-context embedding models. The mentioned studies are evaluated on various retrieval test collections. The concept of interpretability of the embedding vectors is the next topic of this thesis; we show an application of such interpretable representations in gender bias indication of the Wikipedia text. In the last part of the thesis, we investigate the application of our embedding-based methods in two Natural Language Processing (NLP) tasks: CL-WSD for English to Persian, and sentiment analysis of companies' financial reports for volatility prediction of stock markets. In the following, we first summarize each study in turn, followed by discussing the open questions and proposing potential future research direction.

9.1 A Summary of Contributions

In the first study, we address the first research question of the thesis (Q1), namely integrating term associations in the retrieval models. We propose the Generalized and Extended Translation Models, two novel methods to exploit word representations in various models, by expanding the idea of translation model from language modeling to the PR Framework models. The novel translation models assume each query term as a

concept, use the embedding models to find the related terms with common concepts, and finally use this assumption to apply changes in the calculation of core elements of relevance models. We evaluate our models as well as their combinations with Pseudo Relevance Feedback (PRF) on six test collections. The results show a significant improvement of using our approach in comparison to embedding-based query expansion methods, and also the complementary effect of the introduced approaches with PRF.

In the course of the above mentioned study, we observe the importance of selecting the related terms for retrieval performance, and approach it by applying a threshold on the similarity values of the neighboring terms. The optimal value for the threshold is found by a brute-force parameter search. In the next work, as formulated by Q2, we set out to identify the value of such a threshold, by analytically exploring the space of word embedding models. To do it, we first measure the variance of similarity values—referred to as uncertainty—of two arbitrary terms, where the values are determined from different instances of identical embedding models. Using this measure of uncertainty, we then propose the threshold value, estimated based on a novel representation of the neighbors around an arbitrary term. Our evaluation on Ad-hoc retrieval tasks shows that the results using the proposed threshold are either equal to or statistically indistinguishable from the optimal results in the first study. This addresses the second research question of the thesis, namely exploring the range of similarity values as indicative of the actual term relatedness in document retrieval.

We continue the topic of semantic term relatedness in IR, by analyzing the effects of the underlying assumptions in creating embedding models on the sets of related terms. Since word embedding methods use (small) window-context of the terms—as observed in our preliminary experiments—the set of related terms can easily cause topic shifting, when used in document retrieval. This problem is the topic of our third research question Q3: how to enrich window-context word embedding similarities to avoid topic shifting and improve retrieval. To address this question, we study the effectiveness of using the similarities of two embedding models to select the related terms; one based on window-context (word2vec), and the other based on document context (an LSI model created from the term-document matrix). Our evaluation on Ad-hoc retrieval tasks shows the significant improvement of the combined approach. These results demonstrate the importance of considering global context as a complement to the window-context similarities, and motivates future research on learning IR-specific word representation in one embedding space.

The next study investigates our research question, concerning the interpretability of word embedding vectors (Q4). In this work, we propose a novel explicit representation of words (each dimension refers to a term), created based on the word2vec SkipGram model. The proposed representation uses the estimation of the first-order relation in the SkipGram model to create the explicit vectors. The evaluation results on several term association benchmarks show that our explicit SkipGram vectors perform on par with the state-of-the-art explicit representation, confirming the performance of our approach to create explicit representation of the SkipGram model. Further on, we propose a method

using the explicit SkipGram representation to identify and visualize the extent of gender bias, related to a set of occupations in the Wikipedia text. We observe a general tendency of the majority of jobs to the male-related term contexts while strong bias in a few specific occupations to the female-related ones. This observation enables further research, specially on analyzing the effects of such implicit biases on downstream tasks (e.g. job recommendation) as well as algorithmic debiasing of the embedding models.

The next studies investigate our last two questions (Q5 and Q6), regarding the exploitation of the introduced methods in the previous work, in NLP applications. The first one studies the application of semantic-based methods on the CL-WSD task for English to Persian. We propose two unsupervised methods to calculate the semantic similarity of a candidate translation of each ambiguous term, to some translations of the context terms. We use an embedding model in Persian for semantic similarity and a simple bilingual lexicon for achieving the translations. The evaluation of the introduced methods is done on our novel benchmark, created and made available in parallel to the study. Based on the results, our approach outperforms a state-of-the-art system in unsupervised CL-WSD, offering a possible solution for all low-density languages/domains.

In the second application and the last study, we investigate the use of Generalized Translation Models as term weighting schemes for financial sentiment analysis. We explore the benefits of our methods for forecasting volatility, using recent 10-K annual disclosures of companies in stock markets. We also study fusion methods to combine the features of the sentiment analysis method with factual market data. The evaluation shows that our IR-based approach outperforms state-of-the-art volatility prediction methods with 10-K reports and demonstrates the effectiveness of sentiment analysis in long-term volatility forecasting. In addition, we study the extent of the impact of each risk-sensitive term on the sector-specific models, i.e. the models created from the reports of the companies of a specific financial sector. The analysis shows that reports in same sectors considerably share particular risk and instability factors. Despite this fact, due to small size of the available data in each sector, the sector-specific models still do not consistently outperform the general model. This observation highlights the importance of sectors in this context and enables further explorations in the future.

9.2 Open Questions

Regarding the IR topics discussed in the thesis, we see two research areas as open questions and potential future directions.

The first is learning IR-specific word representation from supervised data. In this thesis, specifically in Chapter 5, we highlighted the importance of adapting word embedding for document retrieval tasks. An interesting area for further investigation is learning a neural IR model based on large amount of relevance information, captured specially from abundant log files. A challenging question in this direction is the design of the architectures of such neural approaches in the way that they effectively model the basic components of the classical IR models, discussed in Chapter 3.

The second IR-related direction is learning novel compositional representations as building blocks of neural IR models. We focus in this thesis on word-level representations. An interesting research question is the study of the models to compose document-level representations from word vectors, such as hierarchical neural attention models. In such models, an attention mechanism, conditioned on query and applied on the terms of a section of a document (i.e. a paragraph), first captures the degree of importance of the terms of the section, and then such section-based representations are combined through an hierarchical architecture to compose the final relevance score. This direction is in-line with the study of neural IR models, but also is tightly related to the tasks such as query-based summarization, question answering, and aspect-based sentiment analysis.

We proposed a method to quantify bias in language using explicit word representations. Considering the pervasive use of word embedding in IR and NLP tasks, it is crucial to understand the effects of such bias on downstream tasks. How can bias be measured on task level, and how can it be removed? To what extend does debiasing influence the performance of a system on a specific task? Exploring these questions is indeed an exciting and also crucial research direction.

Finally, the last open question is related to the topic of transfer learning, concerning the task of financial sentiment analysis discussed in Chapter 8. As shown, the financial sectors of the reports play an important role in the performance of sentiment analysis, though the sector-specific models generally suffer from lack of data. Given this issue, how can we design sector-specific models that share a common parameter space for learning the common characteristics among the sectors? To address this question, a potential direction is exploring neural document-level sentiment analysis with parameter sharing.

APPENDIX A

English-Persian Cross-Lingual Word Sense Disambiguation Test Collection

In this appendix, we explain our work on creating a new benchmark for English to Persian Cross-Lingual Word Sense Disambiguation (CL-WSD). In creating the benchmark, we follow the format of the SemEval 2013 CL-WSD task [LH13]. In fact, the new benchmark expands the set of languages of SemEval-2013 CL-WSD task to the Persian language.

We first review the related work and resources for the Persian language in Section A.1, followed by explaining the novel collection in Section A.2.

A.1 Resources in the Persian Language

Persian is a member of the Indo-European language family, and uses Arabic letters for writing. Seraji et al. [SMN12] provide a comprehensive overview of the main characteristics of the language. For instance, the diacritic signs are not written—it is expected that the reader can read the text in the absence of the short vowels. This characteristic causes a special kind of ambiguity in writing, such that some words are pronounced differently while their written forms are the same.

Methods for approaching WSD and CL-WSD highly rely on knowledge and data resources in the language. In the following, we briefly review the main Persian language resources for addressing CL-WSD challenges.

The main knowledge resource in Persian is *FarsNet* [SHF$^+$10]—the Persian WordNet. Its structure is comparable to WordNet and goes by the same principles while containing significantly fewer terms (∼13K versus ∼147K). Also, most of its synsets are mapped

to synsets in WordNet using equal or near-equal relations. Exploiting parallel corpora is another effective method for CL-WSD. In our knowledge, existing parallel corpora (English-Persian) are as follows:

- Tehran English-Persian Parallel (TEP) [PFP11]: a free collection extracted from 1600 movie subtitles.
- Parallel English-Persian News (PEN) [Far11]: the collection aligns 30K sentences of news corpora but is not yet available.
- The collection provided by European Language Resource Association (ELRA) with approximately 100K aligned sentences: ELRA-W0118.

In the absence of reliable and comprehensive resources, some CL-WSD methods exploit the use of a monolingual corpora together with a lexicon. The main available text collections in Persian are Hamshahri [AAD+09], dotIR[1], Bigjekhan[2], and the Uppsala Persian Corpus (UPEC) [SMN12].

In terms of work on WSD and CL-WSD, Saedi et al. [SMS09] exploits the use of WSD in their English-Persian machine translator by first sense disambiguation in the source language and then translating it to the target language. For English-to-Persian translation, they use WordNet in combination with the Lesk algorithm [Les86], while for Persian-to-English, they consider the probability of the co-occurrence of the common senses in a context, learned from a monolingual corpus. More recently, Sarrafzadeh et al. [SYCA11] follow a knowledge-based approach by exploiting FarsNet together with leveraging English sense disambiguation. Their model consists of three phases of: English sense disambiguation, utilizing WordNet and FarsNet to transfer the sense, and selecting the sense from FarsNet. As another method, they investigate direct WSD by applying the extended Lesk algorithm for Persian WSD. They count the number of shared terms between two FarsNet glosses, the gloss of each sense of the target term with the gloss of other terms in the phrases. The one with larger number of common terms is chosen. They test on parallel pages of Wikipedia in English and Persian evaluated by experts. Finally, they show that the first approach works better since they can use the state of the art disambiguator for the English language and the direct approach suffers from lack of NLP tools and ambiguity of Farsi terms.

However, the evaluation resources are not publicly available, which makes it hard to compare their method with other possible approaches. In this work, we address this shortage by creating the new CL-WSD benchmark for Persian, based on the format of the SemEval 2013 CL-WSD task.

[1] http://ece.ut.ac.ir/DBRG/webir/index.html
[2] http://ece.ut.ac.ir/dbrg/Bijankhan

A.2 Persian CL-WSD Evaluation Benchmark

In this section, we describe the process of creating the CL-WSD evaluation benchmark from English to Persian. The novel test collection completely matches the output format of the SemEval 2013 CL-WSD task [LH13] and adds a new language to this multilingual benchmark. In addition, we follow the approach of the original task for creating the gold standard, with only minor alterations necessary in view of the available Persian language resources.

A.2.1 SemEval 2013 CL-WSD

The SemEval 2013 CL-WSD task aims to evaluate the viability of multilingual WSD on a benchmark lexical sample data set. Participants should provide correct translations of English ambiguous nouns, appearing in paragraphs, into five target languages: German, Dutch, French, Italian, and Spanish. The task contains a test set of 20 nouns, each with 50 cases (paragraphs).

Lefever and Hoste [LH13] create the gold standard of the task by first constructing a sense inventory, based on the possible translations of the ambiguous terms. In order to find the target translations, they run term alignment on aligned sentences of the Europarl Corpus [Koe05] and manually verify the results. In the next step, they cluster the resulting translations by meaning per focus terms. Finally, annotators use this clustered sense inventory to select the correct translation for each term, for up to three translations per term.

A.2.2 New Persian Collection

Similar to Lefever and Hoste [LH13], we create our CL-WSD benchmarks in two steps: 1) Creating the sense inventory and 2) Annotation of the translations (ground truth).

In the first step, to create the sense inventory for the 20 nouns, due to the lack of a representative parallel corpora, we leverage three main dictionaries of the Persian language—Aryanpour, Moein, and Farsidic.com—to obtain as large a coverage as possible for their translations. The translations themselves are added by a Persian linguist. In order to provide a thorough set of translations, in addition to different meanings of nouns, their idiomatic meanings (in combinations) are also considered. When the ambiguous word is a part of an idiom (e.g. 'pot' in 'melting pot'), the idiomatic translations are also added to the sense inventory. The number of translations for the terms ranges from 13 to 42, with lowest and highest for the terms 'mood' and 'ring' respectively.

In the next step, the linguist clusters the translations based on their meanings. It results in sense clusters, ranging from 2 to 6 for various nouns. Table A.1 shows the statistics in detail[3].

[3]The sense inventory is available in `https://github.com/navid-rekabsaz/wsd_persian/tree/master/resources/sense-inventory`

Table A.1: The statistics of the English to Persian CL-WSD test collection

Term	# clusters	# translation	% clusters consensus
coach	4	18	98
education	2	15	98
execution	3	14	92
figure	5	33	92
job	3	21	98
letter	4	29	96
match	3	19	96
mission	3	19	98
mood	2	13	100
paper	3	32	98
post	6	38	100
pot	4	34	96
range	5	36	96
rest	4	40	100
ring	6	42	98
scene	4	25	98
side	3	32	96
soil	3	18	100
strain	4	39	98
test	2	13	100

In the second step, the sense inventory is used to annotate the sentences in the test set (50 sentences for each ambiguous term), done by three Persian native-speakers. Via a web-based application, annotators choose the appropriate translations by first selecting the related meaning cluster and then choosing up to three translations from the available list of translations. In case of no related translation, they choose nothing and continue to the next question. Table A.1 shows the number of clusters, number of translation, and the agreement between annotators for selecting the clusters.

Finally, using the annotated data, we create the gold standard in the same format as the SemEval CL-WSD tasks. The gold standard is available in `https://github.com/navid-rekabsaz/wsd_persian/tree/master/resources/golden/Persian`.

APPENDIX B

Gender-Related Terms and Occupations

This appendix contains the full list of female-specific, male-specific, and gender neutral occupations as well as female- and male-specific terms.

Table B.1: Female-specific occupations.

actress	artiste	barmaid	boatwoman	chambermaid
chairwoman	clergywoman	congresswoman	masseuse	midwife
mistress	nanny	draughtwoman	forewoman	furnacewoman
landlady	policewoman	postmistress	postwoman	seawoman
sportwoman	stewardess	stuntwoman	trainwoman	usherette
waitress				

Table B.2: Male-specific occupations.

actor	barman	congressman	clergyman	draughtsman
fisherman	foreman	boatman	chairman	furnaceman
handyman	landlord	masseur	postman	policeman
seaman	sportsman	steward	stuntman	trainman
usher	waiter			

Table B.3: Gender neutral occupations.

accountant	adviser	analyst	animator	announcer
anthropologist	apprentice	archeologist	architect	archivist
artist	assessor	assistant	astrologer	astronomer
athlete	attendant	auctioneer	auditor	bailiff
baker	barber	bargee	bartender	basketmaker
beautician	beekeeper	bibliographer	biochemist	biologist
biotechnologist	blacksmith	boilerfitter	boilermaker	bookbinder
bookkeeper	bookmaker	bootmaker	botanist	breeder
brewer	bricklayer	broadcaster	broker	butcher
butler	buyer	cabinetmaker	captain	caretaker
carpenter	cartographer	cashier	ceramicist	chief
choreographer	cleaner	clerk	coach	collector
commentator	composer	concreter	conductor	confectioner
conservator	consultant	cook	cooper	coremaker
counsellor	courtier	critic	croupier	crusher
curator	cutler	dancer	decorator	dentist
designer	detective	dietician	digger	diplomat
director	dispatcher	diver	doorkeeper	draughtsperson
dresser	dressmaker	driller	drycleaner	dyer
ecologist	economist	editor	educator	electrician
embroiderer	engineer	engraver	environmentalist	ergonomist
ethnographer	expert	farmer	farrier	fitter
furrier	gardener	geneticist	geographer	geologist
geophysicist	gilder	glassmaker	glazier	goatherd
goldsmith	gravedigger	grinder	guard	guide
gunsmith	hairdresser	handler	hardener	harpooner
hatter	healer	herbalist	historian	housekeeper
hydrologist	inspector	instructor	insulator	interpreter
investigator	jeweller	joiner	journalist	judge
knitter	labourer	lacemaker	lawyer	lecturer
librarian	lifeguard	lithographer	maltster	manager
manicurist	master	mathematician	mechanic	melter
merchandiser	metallurgist	metalworker	meteorologist	metrologist
microbiologist	miller	miner	model	modeller
musician	musicologist	naturalist	nurse	nutritionist
obstetrician	officer	operator	optician	optometrist
organizer	orthotist	owner	packer	paediatrician
painter	palmists	paperhanger	paramedic	patternmaker
paver	pawnbroker	pedicurist	performer	pharmacist
philosopher	photographer	physicist	physiotherapist	pilot
planner	plumber	pollster	porter	postmaster
potter	poulterer	priest	producer	programmer
projectionist	prompter	prosecutor	prosthetist	psychiatrist
psychologist	psychotherapist	publisher	radiographer	radiotherapist
receptionist	referee	refiner	registrar	repairer
reporter	representative	rescuer	researcher	retoucher
rigger	roaster	roofer	sausagemaker	scaffolder
scientist	scriptwriter	sculptor	secretary	senior
shepherd	shoemaker	shunter	singer	smith
soldier	solicitor	songwriter	specialist	spinner
staff	statistician	steelworker	steeplejack	stockbroker
stonecutter	stonemason	storekeeper	surgeon	surveyor
sweep	tailor	tamer	tanner	tannery
teacher	technician	technologist	telecaster	teller
therapist	tinsmith	toolmaker	tracklayer	trainer
translator	traveller	tuner	turner	tutor
typesetter	tyrefitter	upholsterer	valuer	varnisher
vendor	viniculturist	warden	washer	weaver
weigher	whaler	wigmaker	worker	zookeeper

Table B.4: Female-specific terms.

daughter	daughters	female	females	fiancee
gal	gals	girl	girls	granddaughter
granddaughters	grandma	grandmother	grandmothers	her
hers	herself	lady	madam	mama
mom	mommy	moms	mother	mothers
she	sister	sisters	stepmother	stepdaughter
woman	women			

Table B.5: Male-specific terms.

boy	boys	brother	brothers	dad
dads	father	fathers	fiance	gentleman
gentlemen	godfather	grandfather	grandpa	grandson
grandsons	guy	he	him	himself
his	lad	lads	male	males
man	men	sir	son	sons
stepfather	stepson			

Table B.6: WinoBias dataset [ZWY+18]. The percent of people in an occupation in the US job market who are reported as female.

Occupation	%	Occupation	%
carpenter	2	editor	52
mechanician	4	designer	54
worker	4	accountant	61
laborer	4	auditor	61
driver	6	writer	63
sheriff	14	baker	65
mover	18	clerk	72
developer	20	cashier	73
farmer	22	counselor	73
guard	22	attendant	76
chief	27	teacher	78
janitor	34	sewer	80
lawyer	35	librarian	84
cook	38	assistant	85
physician	38	cleaner	89
ceo	39	housekeeper	89
analyst	41	nurse	90
manager	43	receptionist	90
supervisor	44	hairdresser	92
salesperson	48	secretary	95

List of Figures

List of Tables

List of Algorithms

Bibliography

[AAD+09] Abolfazl AleAhmad, Hadi Amiri, Ehsan Darrudi, Masoud Rahgozar, and Farhad Oroumchian. Hamshahri: A standard persian text collection. *Knowledge-Based Systems*, 22(5):382–387, 2009.

[AAH+09] Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27. Association for Computational Linguistics, 2009.

[ABK+07] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. *The semantic web*, pages 722–735, 2007.

[ACR04] Giambattista Amati, Claudio Carpineto, and Giovanni Romano. Query difficulty, robustness, and selective application of query expansion. In *Proceeding of the European Conference on Information Retrieval (ECIR)*, pages 127–137. Springer, 2004.

[AVR02] Gianni Amati and Cornelis Joost Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems (TOIS)*, 20(4):357–389, 2002.

[BCZ+16] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 4349–4357, 2016.

[BDK14] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1)*, pages 238–247, 2014.

[BDVJ03] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[BGLM13] Lars Bungum, Björn Gambäck, André Lynum, and Erwin Marsi. Improving word translation disambiguation by capturing multiword expressions with dictionaries. In *MWE workshop at the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 21–30, 2013.

[BL99] Adam Berger and John Lafferty. Information retrieval as statistical translation. In *In Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. Citeseer, 1999.

[BNCB07] Jing Bai, Jian-Yun Nie, Guihong Cao, and Hugues Bouchard. Using query contexts in information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 15–22. ACM, 2007.

[BNJ03] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[Bol86] Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3):307–327, 1986.

[BTB14] Elia Bruni, Nam Khanh Tran, and Marco Baroni. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47, 2014.

[CCD+14] John L Campbell, Hsinchun Chen, Dan S Dhaliwal, Hsin-min Lu, and Logan B Steele. The information content of mandatory risk factor disclosures in corporate filings. *Review of Accounting Studies*, 19(1):396–455, 2014.

[CGS15] Amaru Cuba Gyllensten and Magnus Sahlgren. Navigating the semantic horizon using relative neighborhood graphs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2451–2460, Lisbon, Portugal, 2015.

[CH90] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 22–29, 1990.

[Cho57] Noam Chomsky. *Syntactic Structures*. Mouton and Co., 1957.

[Chu11] Kenneth Church. A pendulum swung too far. *Linguistic Issues in Language Technology*, 6(5):1–27, 2011.

[CLS14] Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1025–1035, 2014.

[CNGR08] Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 243–250. ACM, 2008.

[CSS12] Charlotte Christiansen, Maik Schmeling, and Andreas Schrimpf. A comprehensive look at financial volatility prediction by economic variables. *Journal of Applied Econometrics*, 27(6):956–977, 2012.

[CW08] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pages 160–167. ACM, 2008.

[CWNM02] Hang Cui, Ji-Rong Wen, Jian-Yun Nie, and Wei-Ying Ma. Probabilistic query expansion using query logs. In *Proceedings of the 11th international conference on World Wide Web*, pages 325–332. ACM, 2002.

[DAMR15] Andres Duque, Lourdes Araujo, and Juan Martinez-Romo. Co-graph: A new graph-based technique for cross-lingual word sense disambiguation. *Natural Language Engineering*, FirstView:1–30, 5 2015.

[DDF+90] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391, 1990.

[DL08] Jon Dehdari and Deryle Lonsdale. A link grammar parser for persian. aspects of iranian linguistics, 2008.

[DLSL16] Travis Dyer, Mark H Lang, and Lorien Stice-Lawrence. The ever-expanding 10-k: Why are 10-ks getting so much longer (and does it matter)? *Available at SSRN 2741682*, 2016.

[DMC16] Fernando Diaz, Bhaskar Mitra, and Nick Craswell. Query expansion with locally-trained word embeddings. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 367–377, 2016.

[DMRA15] Andres Duque, Juan Martinez-Romo, and Lourdes Araujo. Choosing the best dictionary for cross-lingual word sense disambiguation. *Knowledge-Based Systems*, 81:65–75, 2015.

[DPL94] Ido Dagan, Fernando Pereira, and Lillian Lee. Similarity-based estimation of word cooccurrence probabilities. In *Proceedings of the annual meeting on Association for Computational Linguistics (ACL)*, pages 272–278. Association for Computational Linguistics, 1994.

[DVZK+14] Lance De Vine, Guido Zuccon, Bevan Koopman, Laurianne Sitbon, and Peter Bruza. Medical semantic similarity with a neural language model. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1819–1822. ACM, 2014.

[DZLD15] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. Deep learning for event-driven stock prediction. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI'15)*, pages 2327–2333, 2015.

[DZS+17] Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W Bruce Croft. Neural ranking models with weak supervision. In *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pages 65–74, 2017.

[Eng82] Robert F Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the Econometric Society*, pages 987–1007, 1982.

[Far11] Mohammad Amin Farajian. Pen: parallel english-persian news corpus. In *Proceedings of the 2011th World Congress in Computer Science, Computer Engineering and Applied Computing*, 2011.

[Fel98] Christiane Fellbaum. *WordNet*. Wiley Online Library, 1998.

[Fir57] J. Firth. A synopsis of linguistic theory 1930-1955. In *Studies in Linguistic Analysis*. Philological Society, Oxford, 1957.

[FTY+15] Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah A Smith. Sparse overcomplete word vector representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 1, pages 1491–1500, 2015.

[FZ06] Hui Fang and ChengXiang Zhai. Semantic term matching in axiomatic approaches to information retrieval. In *Proceedings of the International ACM SIGIR conference on Research and development in information retrieval*, pages 115–122. ACM, 2006.

[GA11] Mehmet Gönen and Ethem Alpaydın. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12(Jul):2211–2268, 2011.

[GDR+15] Mihajlo Grbovic, Nemanja Djuric, Vladan Radosavljevic, Fabrizio Silvestri, and Narayan Bhamidipati. Context-and content-aware embeddings for query rewriting in sponsored search. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 383–392. ACM, 2015.

[GFAC16] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 55–64. ACM, 2016.

[GN12] Jianfeng Gao and Jian-Yun Nie. Towards concept-based translation models using search logs for query expansion. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, page 1. ACM, 2012.

[GRMJ15] Debasis Ganguly, Dwaipayan Roy, Mandar Mitra, and Gareth JF Jones. Word embedding based generalized language model for information retrieval. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 795–798. ACM, 2015.

[Har51] Zellig S Harris. Methods in structural linguistics. 1951.

[HC14] Samuel Huston and W Bruce Croft. A comparison of retrieval models using term dependencies. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 111–120. ACM, 2014.

[Hof99] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

[HRK15] Felix Hill, Roi Reichart, and Anna Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695, 2015.

[HSMN12] Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882, 2012.

[JMD10] Amir Hossein Jadidinejad, Fariborz Mahmoudi, and Jon Dehdari. Evaluation of perstem: a simple and efficient stemming algorithm for persian. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments*, pages 98–101. Springer, 2010.

[Jon03] Karen Sparck Jones. Document retrieval: Shallow data, deep theories; historical reflections, potential directions. *Lecture notes in computer science*, pages 1–11, 2003.

[KB15] Germán Kruszewski and Marco Baroni. So similar and yet incompatible: Toward the automated identification of semantically compatible words. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 964–969, 2015.

[KBE+14] Jussi Karlgren, Martin Bohman, Ariel Ekgren, Gabriel Isheden, Emelie Kullmann, and David Nilsson. Semantic topology. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1939–1942. ACM, 2014.

[KHC15] Douwe Kiela, Felix Hill, and Stephen Clark. Specializing word embeddings for similarity or relatedness. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2044–2048, 2015.

[KHS08] Jussi Karlgren, Anders Holst, and Magnus Sahlgren. Filaments of meaning in word space. In *Proceedings of European Conference on Information Retrieval (ECIR)*, pages 531–538. Springer, 2008.

[KLR+09] Shimon Kogan, Dimitry Levin, Bryan R Routledge, Jacob S Sagi, and Noah A Smith. Predicting risk from financial reports with regression. In *Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 272–280, 2009.

[Koe05] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*, 2005.

[KSKW15] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966, 2015.

[KZ10] Maryam Karimzadehgan and ChengXiang Zhai. Estimation of statistical translation models based on mutual information for ad hoc information retrieval. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 323–330. ACM, 2010.

[KZ12a] Maryam Karimzadehgan and ChengXiang Zhai. Axiomatic analysis of translation language model for information retrieval. *Advances in Information Retrieval*, pages 268–280, 2012.

[KZ12b] Alexander Kotov and ChengXiang Zhai. Tapping into knowledge base for concept feedback: leveraging conceptnet to improve search results for

difficult queries. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 403–412. ACM, 2012.

[KZB+12] Bevan Koopman, Guido Zuccon, Peter Bruza, Laurianne Sitbon, and Michael Lawley. An evaluation of corpus-driven measures of medical concept similarity for information retrieval. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2439–2442. ACM, 2012.

[KZP14] Siavash Kazemian, Shunan Zhao, and Gerald Penn. Evaluating sentiment analysis evaluation: A case study in securities trading. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, page 119, 2014.

[L+09] Tie-Yan Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.

[Lac96] A. R. Lacey. *A dictionary of philosophy*. Routledge London, 1996.

[LC01] Victor Lavrenko and W Bruce Croft. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 120–127. ACM, 2001.

[LCB+04] Gert RG Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine learning research*, 5(Jan):27–72, 2004.

[Ld15] Ronny Luss and Alexandre d'Aspremont. Predicting abnormal returns from news using text classification. *Quantitative Finance*, 15(6):999–1012, 2015.

[Les86] Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26. ACM, 1986.

[LG14] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2177–2185, 2014.

[LGD15] Omer Levy, Yoav Goldberg, and Ido Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015.

[LH10] Els Lefever and Veronique Hoste. Semeval-2010 task 3: Cross-lingual word sense disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 15–20. Association for Computational Linguistics, 2010.

[LH11] Hongquan Li and Yongmiao Hong. Financial volatility forecasting with range-based autoregressive volatility model. *Finance Research Letters*, 8(2):69–76, 2011.

[LH13] Els Lefever and Veronique Hoste. Semeval-2013 task 10: cross-lingual word sense disambiguation. In *7th International workshop on Semantic Evaluation (SemEval 2013)*, pages 158–166. Association for Computational Linguistics (ACL), 2013.

[LHDC11] Els Lefever, Véronique Hoste, and Martine De Cock. Parasense or how to use parallel corpora for word sense disambiguation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 317–322. Association for Computational Linguistics, 2011.

[Li10] Feng Li. The information content of forward-looking statements in corporate filings–a naïve bayesian machine learning approach. *Journal of Accounting Research*, 48(5):1049–1102, 2010.

[LLHA15] Aldo Lipani, Mihai Lupu, Allan Hanbury, and Akiko Aizawa. Verboseness fission for bm25 document length normalization. In *Proc. of International Conference on the Theory of Information Retrieval (ICTIR)*, 2015.

[LM11] Tim Loughran and Bill McDonald. When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *The Journal of Finance*, 66(1):35–65, 2011.

[LSLH15] Christina Lioma, Jakob Grue Simonsen, Birger Larsen, and Niels Dalum Hansen. Non-compositional term dependence for information retrieval. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 595–604. ACM, 2015.

[LSM13] Thang Luong, Richard Socher, and Christopher D Manning. Better word representations with recursive neural networks for morphology. In *In Proceedings of the Computational Natural Language Learning Conference (CoNLL)*, pages 104–113, 2013.

[LT13] Shouwei Liu and Yiu Kuen Tse. Estimation of monthly volatility: An empirical comparison of realized volatility, garch and acd-icv methods. *Research Collection School Of Economics*, 2013.

[LX14] Hang Li and Jun Xu. Semantic Matching in Search. *Foundations and Trends in Information Retrieval*, 2014.

[LZ03] John Lafferty and Chengxiang Zhai. Probabilistic relevance models based on document and query generation. In *Language modeling for information retrieval*, pages 1–10. Springer, 2003.

[LZ09] Yuanhua Lv and ChengXiang Zhai. Positional language models for information retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 299–306. ACM, 2009.

[MCCD13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[MDC17] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1291–1299. International World Wide Web Conferences Steering Committee, 2017.

[Mit15] Bhaskar Mitra. Exploring session context using distributed representations of queries and reformulations. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12. ACM, 2015.

[MP69] Marivn Minsky and Seymour Papert. *Perceptrons*. MIT Press, 1969.

[MSC+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, 2013.

[MT12] Andriy Mnih and Yee W Teh. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1751–1758, 2012.

[MVN13] Rodrigo Moraes, JoãO Francisco Valiati, and Wilson P GaviãO Neto. Document-level sentiment classification: An empirical comparison between svm and ann. *Expert Systems with Applications*, 40(2):621–633, 2013.

[NH15] Clemens Nopp and Allan Hanbury. Detecting risks in the banking system by sentiment analysis. *Proceedings of the Conference of Empirical Methods in Natural Language Processing (EMNLP)*, pages 591–600, 2015.

[NN94] Yoshiki Niwa and Yoshihiko Nitta. Co-occurrence vectors from corpora vs. distance vectors from dictionaries. In *Proceedings of the annual meeting on Association for Computational Linguistics (ACL)*, pages 304–309, 1994.

[Nob04] William Stafford Noble. Support vector machine applications in computational biology. *Kernel methods in computational biology*, pages 71–92, 2004.

[NP10] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting*

of the association for computational linguistics, pages 216–225. Association for Computational Linguistics, 2010.

[NS15] Thien Hai Nguyen and Kiyoaki Shirai. Topic modeling based sentiment analysis on social media for stock market prediction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1354–1364, 2015.

[Pai13] Jiaul H Paik. A novel tf-idf weighting scheme for effective ranking. In *Proceedings of the international ACM SIGIR conference on Research and development in information retrieval*, pages 343–352. ACM, 2013.

[PC98] Jay M Ponte and W Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281. ACM, 1998.

[PFP11] Mohammad Taher Pilevar, Heshaam Faili, and Abdol Hamid Pilevar. Tep: Tehran english-persian parallel corpus. In *Computational Linguistics and Intelligent Text Processing*, pages 68–79. Springer, 2011.

[PLSL15] Casper Petersen, Christina Lioma, Jakob Grue Simonsen, and Birger Larsen. Entropy and graph based modelling of document coherence using discourse entities: An application to ir. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*, pages 191–200. ACM, 2015.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[PW91] Helen J Peat and Peter Willett. The limitations of term co-occurrence data for query expansion in document retrieval systems. *Journal of the American society for information science*, 1991.

[PZG+15] João RM Palotti, Guido Zuccon, Lorraine Goeuriot, Liadh Kelly, Allan Hanbury, Gareth JF Jones, Mihai Lupu, and Pavel Pecina. Clef ehealth evaluation lab 2015, task 2: Retrieving information about medical symptoms. In *Proceedings of Conference and Labs of the Evaluation Forum (CLEF)*, 2015.

[RBI+15] Navid Rekabsaz, Ralf Bierig, Bogdan Ionescu, Allan Hanbury, and Mihai Lupu. On the use of statistical semantics for metadata-based social image retrieval. In *Content-Based Multimedia Indexing (CBMI), 2015 13th International Workshop on*, pages 1–4. IEEE, 2015.

[RBLH15] Navid Rekabsaz, Ralf Bierig, Mihai Lupu, and Allan Hanbury. Toward optimized multimodal concept indexing. In *Proceedings of the International KEYSTONE Conference on Semantic Keyword-based Search on Structured Data Sources (IKC)*, pages 141–152. Springer, 2015.

[RBLH17] Navid Rekabsaz, Ralf Bierig, Mihai Lupu, and Allan Hanbury. Toward optimized multimodal concept indexing. In *Transactions on Computational Collective Intelligence XXVI*, pages 144–161. Springer, 2017.

[RJSK10] Gabriel Recchia, Michael Jones, Magnus Sahlgren, and Pentti Kanerva. Encoding sequential information in vector space models of semantics: Comparing holographic reduced representation and random permutation. In *Proceedings of the Cognitive Science Society*, volume 32, 2010.

[RLB+17] Navid Rekabsaz, Mihai Lupu, Artem Baklanov, Alexander Dür, Linda Andersson, and Allan Hanbury. Volatility prediction using financial disclosures sentiments with word embedding-based ir models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1712–1721, 2017.

[RLG13] Alex Rudnick, Can Liu, and Michael Gasser. Hltdi: Cl-wsd using markov random fields for semeval-2013 task 10. In *Second Joint Conference on Lexical and Computational Semantics (* SEM)*, volume 2, pages 171–177, 2013.

[RLH16] Navid Rekabsaz, Mihai Lupu, and Allan Hanbury. Uncertainty in neural network word embedding: Exploration of threshold for similarity. *NeuIR Workshop at the ACM Conference on Research and Development in Information Retrieval (SIGIR)*, 2016.

[RLH17] Navid Rekabsaz, Mihai Lupu, and Allan Hanbury. Exploration of a threshold for similarity based on uncertainty in word embedding. In *European Conference on Information Retrieval*, pages 396–409. Springer, 2017.

[RLHD17] Navid Rekabsaz, Mihai Lupu, Allan Hanbury, and Andres Duque. Addressing cross-lingual word sense disambiguation on low-density languages: Application to persian. *arXiv preprint arXiv:1711.06196*, 2017.

[RLHZ16] Navid Rekabsaz, Mihai Lupu, Allan Hanbury, and Guido Zuccon. Generalizing translation models in the probabilistic relevance framework. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 711–720. ACM, 2016.

[RLHZ17] Navid Rekabsaz, Mihai Lupu, Allan Hanbury, and Hamed Zamani. Word embedding causes topic shifting; exploit global context! In *Proceedings of the International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1105–1108. ACM, 2017.

[RMLH17] Navid Rekabsaz, Bhaskar Mitra, Mihai Lupu, and Allan Hanbury. Toward incorporation of relevant documents in word2vec. *Neu-IR Workshop at the ACM Conference on Research and Development in Information Retrieval (SIGIR)*, 2017.

[Rob05] Stephen Robertson. On Event Spaces and Probabilistic Models in Information Retrieval. *Information Retrieval*, 8, 2005.

[Roc71] J.J. Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System– Experiments in Automatic Document Processing*, 1971.

[ŘS10] R. Řehůřek and P. Sojka. Software framework for topic modelling with large corpora. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC) - Workshop on New Challenges for NLP*, 2010.

[RSL+16] Navid Rekabsaz, Serwah Sabetghadam, Mihai Lupu, Linda Andersson, and Allan Hanbury. Standard test collection for english-persian cross-lingual word sense disambiguation. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2016.

[RZ+09] Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

[Sah05] Magnus Sahlgren. An introduction to random indexing. In *Methods and Applications of Semantic Indexing Workshop in the Proceeding of TKE Conference*, volume 5, 2005.

[Sak07] Tetsuya Sakai. Alternatives to bpref. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 71–78. ACM, 2007.

[Sal68] Gerard Salton. Automatic information organization and retrieval. *New York: McGraw-Hill*, 1968.

[SBM96] Amit Singhal, Chris Buckley, and Mandar Mitra. Pivoted document length normalization. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–29, 1996.

[SC16] Martin Stražar and Tomaž Curk. Learning the kernel matrix via predictive low-rank approximations. *arXiv preprint arXiv:1601.04366*, 2016.

[Sch92] Hinrich Schutze. Dimensions of meaning. In *Supercomputing '92*, pages 787–796. IEEE, 1992.

[SGL+16] Fei Sun, Jiafeng Guo, Yanyan Lan, Jun Xu, and Xueqi Cheng. Sparse word embeddings using l1 regularized online learning. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

[Sha48] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

[SHF+10] Mehrnoush Shamsfard, Akbar Hesabi, Hakimeh Fadaei, Niloofar Mansoory, Ali Famian, Somayeh Bagherbeigi, Elham Fekri, Maliheh Monshizadeh, and S Mostafa Assi. Semi automatic development of farsnet; the persian wordnet. In *Proceedings of 5th Global WordNet Conference, Mumbai, India*, 2010.

[SLMJ15] Tobias Schnabel, Igor Labutov, David M Mimno, and Thorsten Joachims. Evaluation methods for unsupervised word embeddings. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 298–307, 2015.

[SM83] G Salton and MJ MacGill. Introduction to modern information retrieval. *McGraw-Hill*, 1983.

[SM15] Aliaksei Severyn and Alessandro Moschitti. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 373–382. ACM, 2015.

[SMN12] Mojgan Seraji, Beáta Megyesi, and Joakim Nivre. A basic language resource kit for persian. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 2245–2252, 2012.

[SMS09] Chakaveh Saedi, Yasaman Motazadi, and Mehrnoush Shamsfard. Automatic translation between english and persian texts. In *Proceedings of the Third Workshop on Computational Approaches to Arabic Script-based Languages, Ottawa, Ontario, Canada*, 2009.

[SP93] Hinrich Schütze and Jan Pedersen. A vector model for syntagmatic and paradigmatic relatedness. In *Proceedings of the 9th Annual Conference of the UW Centre for the New OED and Text Research*, pages 104–113. Oxford, 1993.

[Ste11] Mark Steedman. Romantics and revolutionaries. *Linguistic Issues in Language Technology*, 6(11):1–20, 2011.

[SVT12] Sokratis Sofianopoulos, Marina Vassiliou, and George Tambouratzis. Implementing a language-independent mt methodology. In *Proceedings of the First Workshop on Multilingual Modeling*, pages 1–10, 2012.

[SYCA11] Bahareh Sarrafzadeh, Nikolay Yakovets, Nick Cercone, and Aijun An. Cross-lingual word sense disambiguation for languages with scarce resources. In *Advances in Artificial Intelligence: Proceedings of 24th Canadian Conference on Artificial Intelligence*, pages 347–358. Springer Berlin Heidelberg, 2011.

[TFL+15] Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. Evaluation of word vector representations by subspace alignment. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

[TLLH14] Xinhui Tu, Jing Luo, Bo Li, and Tingting He. Log-bilinear document language model for ad-hoc information retrieval. In *Proceedings of the ACM International Conference on Conference on Information and Knowledge Management*, pages 1895–1898. ACM, 2014.

[TW14] Ming-Feng Tsai and Chuan-Ju Wang. Financial keyword expansion via continuous word vector representations. In *Proceedings of the Conference of Empirical Methods in Natural Language Processing (EMNLP)*, pages 1453–1458, 2014.

[TZ07] Tao Tao and ChengXiang Zhai. An exploration of proximity measures in information retrieval. In *Proceedings of the annual international ACM SIGIR conference on Research and development in information retrieval*, pages 295–302, 2007.

[VM15] Ivan Vulić and Marie-Francine Moens. Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 363–372. ACM, 2015.

[WC06] Xing Wei and W Bruce Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the International ACM SIGIR conference on Research and development in information retrieval*, pages 178–185. ACM, 2006.

[WH14] William Yang Wang and Zhenhao Hua. A semiparametric gaussian copula regression model for predicting financial risks from earnings calls. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1155–1165, 2014.

[Wil08] Yorick Wilks. What would a Wittgensteinian computational linguistics be like? In *Convention Communication, Interaction and Social Intelligence (AISB)*, volume 1, page 1, 2008.

[Wil11] Yorick Wilks. Computational semantics requires computation. *Cross-Disciplinary Advances in Applied Natural Language Processing: Issues and Approaches*, pages 1–8, 2011.

[Wit53] Ludwig Wittgenstein. *Philosophical investigations. Philosophische Untersuchungen.* Macmillan, 1953.

[Wol92] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[WTLC13] Chuan-Ju Wang, Ming-Feng Tsai, Tse Liu, and Chin-Ting Chang. Financial sentiment analysis for risk prediction. In *Proceedings of the Joint Conference on Natural Language Processing (IJCNLP)*, pages 802–808, 2013.

[XC96] Jinxi Xu and W Bruce Croft. Query expansion using local and global document analysis. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1996.

[XC15] Chenyan Xiong and Jamie Callan. Query expansion with freebase. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*, pages 111–120. ACM, 2015.

[XDC+17] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 55–64. ACM, 2017.

[XJW09] Yang Xu, Gareth JF Jones, and Bin Wang. Query dependent pseudo-relevance feedback based on wikipedia. In *Proceedings of the International ACM SIGIR Conference on Research and development in information retrieval*, pages 59–66. ACM, 2009.

[XPW13] Boyi Xie, Rebecca J Passonneau, and Leon Wu. Semantic Frames to Predict Stock Price Movement. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 2013.

[ZC15] Guoqing Zheng and Jamie Callan. Learning to reweight terms with distributed representations. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 575–584. ACM, 2015.

[ZC16] Hamed Zamani and W Bruce Croft. Embedding-based query language models. In *Proceedings of the 2016 ACM on International Conference on the Theory of Information Retrieval*, pages 147–156. ACM, 2016.

[ZC17] Hamed Zamani and W Bruce Croft. Relevance-based word embedding. In *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)*, 2017.

[ZHY14] Jiashu Zhao, Jimmy Xiangji Huang, and Zheng Ye. Modeling term associations for probabilistic information retrieval. *ACM Transactions on Information Systems (TOIS)*, 32(2):7, 2014.

[ZKBA15] Guido Zuccon, Bevan Koopman, Peter Bruza, and Leif Azzopardi. Integrating and evaluating neural word embeddings in information retrieval. In *Proceedings of the 20th Australasian Document Computing Symposium*, page 12. ACM, 2015.

[ZWY+18] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, volume 2, pages 15–20, 2018.