UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Word Representation with Transferable Semantics

by Qian Liu

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of Distinguished Prof. Jie Lu and Associate Prof. Guangquan Zhang

University of Technology Sydney
Faculty of Engineering and Information Technology

May 2021

Certificate of Original Authorship Template

Graduate research students are required to make a declaration of original authorship when they submit the thesis for examination and in the final bound copies. Please note, the Research Training Program (RTP) statement is for all students. The Certificate of Original Authorship must be placed within the thesis, immediately after the thesis title page.

Required wording for the certificate of original authorship

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, *Qian Liu* declare that this thesis, is submitted in fulfilment of the requirements for the award of *Doctor of Philosophy*, in the *Faculty of Engineering and Information Technology* at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.
**If applicable, the above statement must be replaced with the collaborative doctoral degree statement (see below).*

**If applicable, the Indigenous Cultural and Intellectual Property (ICIP) statement must be added (see below).*

This research is supported by the Australian Government Research Training Program.

Signature: Production Note:
Signature removed prior to publication.

Date: 18/05/2021

Collaborative doctoral research degree statement

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of the requirements for a degree at any other academic institution except as fully acknowledged within the text. This thesis is the result of a Collaborative Doctoral Research Degree program with *Beijing Institute of Technology*.

Indigenous Cultural and Intellectual Property (ICIP) statement

This thesis includes Indigenous Cultural and Intellectual Property (ICIP) belonging to *N/A*, custodians or traditional owners. Where I have used ICIP, I have followed the relevant protocols and consulted with appropriate Indigenous people/communities about its inclusion in my thesis. ICIP rights are Indigenous heritage and will always remain with these groups. To use, adapt or reference the ICIP contained in this work, you will need to consult with the relevant Indigenous groups and follow cultural protocols.

Abstract

Semantic representation aims to encode the meaning of text (e.g., words) in a form which can be stored and processed by a machine, such as real-valued vectors or neural networks with well-trained parameters. In particular, semantic knowledge is expected to be embodied in representations. For example, words with similar meanings are expected to be close to each other when they are represented as vectors. Semantic representation is the basic block of neural networks, and it should have better expression ability to support downstream natural language processing applications.

Although recent research on semantic representation has shown a reasonable ability to represent textual data by only using large-scale raw text, most research is incomplete and biased as it only models the surface co-occurrence information of corpora but ignores deep semantic and syntactic information. In addition, most research focuses on modeling generic semantics, while disengaging from task requirements. Hence, existing semantic representation methods still face several unsolved and challenging problems in the real world.

This thesis aims to design better representation learning methods by utilizing *transferable* semantics extracted from source domains, which are resourceful and beyond raw text. More specifically, this thesis aims to address four problems faced by existing semantic representation methods: 1) how to reliably transfer semantics from a structural knowledge base to an unstructured representation space; 2) how to reliably transfer semantics from multiple source domains to a low-resource target domain; 3) how to achieve the reliable and low-cost cross-lingual transfer of semantics; and 4) how to adapt semantic representations for specific applications.

To address Problem 1), this thesis designs two assumptions to model semantic structures in knowledge bases and proposes a new semantic structure-based semantic representation method (Chapter 3). It leverages the human-defined relationships among words from structured knowledge bases as transferable semantics to improve its representation ability. Instead of using the relations between word-pairs, our method uses whole semantic structures which have proven to be more effective in semantic representation.

To address Problem 2), this thesis proposes a dynamical meta-embedding method to leverage the semantics from multiple source domains (Chapter 4). It leverages latent knowledge from multiple source embeddings to improve representation learning for a low-resource domain. Considering domain shifts and quality discrepancy, it dynamically aggregates multiple source embeddings by a differentiable attention module, instead of using them equally. It is proven to be more suitable to transfer true required semantics from multiple source domains to a low-resource domain.

To address Problem 3), this thesis proposes a new method to bridge the cross-lingual semantic gap with limited bilingual resource reliance (Chapter 5). Based on multilingual embeddings, it learns a pivot set which is semantically related to a low-resource language and lexically related to a high-resource language. With the learned pivots, our method is useful to help models trained on high-resource languages to be adapted on low-resource languages.

To address Problem 4), this thesis proposes a fuzzy word similarity measure to adapt general semantic representations according to the need of a specific task (Chapter 6). It takes task-oriented features into consideration and adapts general semantics to the specific tasks, which alleviates the problem of disengaging from task requirements.

To conclude, this thesis proposes a set of effective methods to improve semantic representation by exploring and modeling knowledge beyond raw text and places an emphasis on encoding task-specific features for real-world applications.

ACKNOWLEDGMENTS

It is an exciting journey at University of Technology Sydney (UTS) for pursuing my Ph.D. degree in the past four years. I am sincerely grateful to the people who inspired and helped me in many ways. I would like to express my foremost and deepest gratitude to my principal supervisor, Distinguished Professor Jie Lu. Her decisiveness and sharp insights continuously motivated me when I got lost or afraid about the future. Her confidence and enthusiasm inspired me to do the right thing even when the road got tough. She placed considerable trust in my research ability and unconditionally support me in pursuing my own research interests. Her wisdom and immense knowledge always enlightened me to go further and deeper in my research. I felt extremely honored to be guided by such a rigorous researcher as well as an enthusiastic mentor. What she taught me and what I learned from her in the past four years has benefited my Ph.D. study and will be a great treasure throughout my life.

Meanwhile, I am greatly indebted to my co-advisor, A./Professor Guangquan Zhang. Without his patience and encouragement, I would not have been able to complete this Ph.D. program. He taught me step by step how to become a qualified researcher from its beginning. He always led me to the right research direction with his expert knowledge of theory and abundant research experience. Without his critical comments, I would waste my time on trivial research ideas. Discussion with him greatly improves the scientific aspect and quality of my research. He helped me to build my confidence in my research outcomes and to be hopeful when faced with any difficulty, from academic to living.

I would like to express my thankfulness to every member of the Decision Systems & e-Service Intelligence Lab (DeSI) in the Australian Artificial Intelligence Institute (AAII).

It was a wonderful experience to spend four years with these dedicated researchers. I especially thank Dr. Junyu Xuan, Dr. Tao Shen, Dr. Yi Zhang, Dr. Fan Dong, Dr. Feng Gu, Dr. Anjin Liu and Dr. Hua Zuo who provided insightful comments related to my research problem during my Ph.D. candidature; Dr. Ruiping Yin, Dr. Hang Yu, Dr. Guanjin Wang, Dr. Chenlian Hu, Dr. Feng Liu, Dr. Yiliao Song, Dr. Adi Lin, and Bin Wang who have shared their opinions and comments with me; Dr. Dan Shang, Dr. Shan Xue, Dr. Guanjin Wang, and Dr. Xiaohang Xu, who shared my joys and sadness.

Last, I would like to express my heartfelt appreciation and gratitude to my parents and families for their love and support.

List of Publications

1. **Qian Liu**, Xiubo Geng, Jie Lu, Daxin Jiang. Pivot-based Candidate Retrieval for Cross-lingual Entity Linking. *Proceedings of the 30th The Web Conference (WWW-21)*, Ljubljana, Slovenia, April 12-16, 2021. [ERA: A, CORE: A*]

2. **Qian Liu**, Xiubo Geng, Tao Qin, Heyan Huang, Jie Lu, Daxin Jiang. MGRC: An End-to-End Multi-Granularity Reading Comprehension Model for Question Answering. *IEEE Transactions on Neural Networks and Learning Systems (IEEE-TNNLS)*, 2021. [ERA&CORE: A*, JCR Q1]

3. **Qian Liu**, Jie Lu, Guangquan Zhang, Tao Shen, Zhihan Zhang, Heyan Huang. Domain-specific meta-embedding with latent semantic structures. *Information Sciences*, Vol. 555, pp. 410-423, May 2021. DOI: 10.1016/j.ins.2020.10.030. [CORE: A, JCR Q1]

4. **Qian Liu**, Heyan Huang, Junyu Xuan, Guangquan Zhang, Jie Lu. A Fuzzy Word Similarity Measure for Selecting Top-k Similar Words for Query Expansion. *IEEE Transactions on Fuzzy Systems (IEEE-TFS)*, 2020. DOI: 10.1109/T-FUZZ.2020.2993702. [ERA&CORE: A*, JCR Q1]

5. **Qian Liu**, Heyan Huang, Guangquan Zhang, Yang Gao, Junyu Xuan, Jie Lu. Semantic Structure based Word Embedding by Incorporating Concept Convergence and Word Divergence. *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, pp.5261-5268, New Orleans, LA, USA, 2018. [ERA: A, CORE: A*]

6. **Qian Liu**, Heyan Huang, Jie Lu, Yang Gao, Guangquan Zhang. Enhanced Word Embedding Similarity Measures using Fuzzy Rules for Query Expansion. *Proceedings of the 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE-17)*, pp. 1-6, Naples, Italy, 9-12 July, 2017. [ERA&CORE: A]

7. **Qian Liu**, Jie Lu, Guangquan Zhang. Towards Zero-shot Cross-lingual Entity Linking. *IEEE Transactions on Knowledge and Data Engineering (IEEE-TKDE)*, 2021. [CORE: A*, ERA: A, JCR Q1] (Under Review)

8. Yi Zhang, Jie Lu, Feng Liu, **Qian Liu**, Alan Porter, Hongshu Chen, Guangquan Zhang. Does deep learning help topic extraction? A kernel k-means clustering method with word embedding. *Journal of Informetrics*, Vol. 12, no. 4, pp. 1099-1117, 2018. [ERA&CORE: A, JCR Q1]

   Note: Chapter 3 relates paper 5, Chapter 4 relates paper 3, Chapter 5 relates paper 1, and Chapter 6 relates paper 4.

TABLE OF CONTENTS

List of Figures

List of Tables

Table **Page**

CHAPTER 1

INTRODUCTION

1.1 Background

Understanding the meaning of a word and extending that of larger units (e.g., phrase, sentence, and paragraph) is the core research problem of text-based machine learning, e.g., natural language processing (NLP) [1] and information retrieval (IR) [2, 3]. To gain a deep understanding of textual data, it is necessary to represent it in a form which computers can store and in a form on which they can operate. How to represent words in a way that embeds as much semantics as possible is the fundamental challenge.

According to Antony and Davies [4], *semantic knowledge* is what a speaker knows in knowing their own languages. Modeling semantic knowledge in machine learning methods is not trivial. A popular hypothesis to model semantic knowledge is that *words that occur in a similar context tend to have similar meanings* [5], which is also known as *distributed hypothesis*. Based on this hypothesis and with the help of rapidly developing neural networks, researchers have deigned the first generation of semantic representation called *distributed embeddings*, which represent words as low-dimension vectors. These methods can be divided into two categories: (1) count-based methods

which compute the statistics of how often some words co-occur with their neighbor words in a large text corpus, and then map these count statistics down to a small, dense vector for each word [6, 7]; (2) prediction-based methods which directly try to predict a word from its neighbors in terms of learned small, dense embedding vectors (considered parameters of the model), such as neural probabilistic language models [8]. Despite their effectiveness, distributed embeddings suffer from the out-of-vocabulary problem and the word disambiguation problem since they represent each word as a fixed vector regardless of its context. More recently, the second generation of semantic representation called *pre-trained language model* has been devised, for example ELMo [9], BERT [10], RoBERTa [11] and XLNet [12]. They first pre-train deep language models on large-scale unlabeled text corpora, and then fine-tune the models or representations on downstream tasks. With a powerful ability to capture the meaning of words from the discourse context, they have led to significant performance gain and achieve state-of-the-art for a wide range of applications.

Although it is attractive to learn semantic representations purely from raw corpora, they are not the only source of semantic knowledge. Subsequently, researchers have considered how to leverage the knowledge beyond corpora to help improve semantic representations, for example, the knowledge gained from semantic lexicons (e.g., WordNet [13]) and commonsense knowledge bases (e.g., ConceptNet [14]), rich semantic information that exist in the corpus (e.g., topic [15] and associated patterns [16]) or tasks (e.g., category information for text classification [17] and affective polarity for sentiment analysis [18]).

To leverage knowledge from other resources, transfer learning methods [19] have demonstrated good success in various practical applications in recent years. Typical transfer learning aims to leverage knowledge from domains with abundant labels (i.e., source domains) to help train a classifier or predictor for the domain with insufficient labels (i.e., target domain). For representation learning of textual data, we aim to leverage knowledge from source domains with rich semantic information to help train better semantic representations for the target domain with limited resources. The learned

representations can better support the tasks defined in the target domain. The used semantic knowledge in the source domain is denoted as *transferable semantics* in this thesis. In the real world, the knowledge of semantics in different domains is latent and heterogeneous. In this thesis, we need to study how to extract, model, and transfer semantic knowledge from various resources beyond raw corpora to learn better semantic representations[1].

1.2 Motivation

The dilemma in leveraging semantic knowledge from source domains beyond raw corpora is mainly due to its two inherent features. First, it is *latent*. Semantic knowledge exists implicitly in resources. It is necessary to design reasonable theories to discover and model semantic knowledge. Second, it is *heterogeneous* across domains. For example, considering that the knowledge of "the concept *animal* contains *dog*", it is represented as a sentence *"a dog is an animal"* in a corpus, or a triplet <*dog, isA, animal*> in a knowledge graph. To transfer semantic knowledge across domains, it is necessary to design a joint learning method to capture heterogeneous knowledge into a unity representation space. To go a step further, semantic representation is learned to support downstream tasks. It is also important to emphasize *task-specific* knowledge in the representation space, for example, capturing the sentiment features of words for sentiment analysis.

This thesis finds four unsolved challenges faced by existing methods and proposes corresponding methods to address these challenges. The four challenges are 1) how to reliably transfer semantics from a structural knowledge base to an unstructured representation space; 2) how to reliably transfer semantics from multiple source domains to a low-resource target domain; 3) how to achieve the reliable and low-cost cross-lingual transfer of semantics; and 4) how to adapt semantic representations for specific applica-

[1]In this thesis, we primarily considered *distributed representation methods* since 1) they are widely used on NLP and IR tasks, and they still occupy a dominant position in large-scale applications; 2) another approach (i.e., pre-trained language models) consumes huge resources and depends on industrial-grade computing power (e.g., GPUs).

tions. This thesis gives a comprehensive analysis and solutions to all the aforementioned challenges.

1.3 Research Questions and Objectives

This thesis develops a set of representation learning methods using transferable semantics and answers the following research questions:

Research Question 1 (RQ1): *how to reliably transfer semantics from a structural knowledge base to an unstructured representation space?*

Representing the semantics of text is a fundamental task in natural language processing. With the underlying idea that "a word is characterized by the company it keeps" [5], researchers have developed many neural networks to encode the semantics of linguistic items (such as words, terms, and sentences) based on their distributional properties in large-scale textual data. For example, *prediction-based* methods [8, 20] which learn semantic representation by predicting the co-occurrence of words in the given context, and *counting-based* methods [7] which learn word representations through global matrix factorization based on a count of co-occurring words. In the real world, some human-crafted knowledge bases (such as WordNet [13] and ConceptNet [14]) contain well-organized structural information of words which can convey effective and stable knowledge in capturing the semantics of words. Therefore, unstructured textual data and structural knowledge bases are complementary sources which learn high-quality semantic representations. However, most of the research directs attention entirely towards learning semantic representations from a large-scale corpus, ignoring the valuable semantic structures in knowledge bases. To jointly model semantics in knowledge bases and the corpus, the main obstacle is that *knowledge* and *contextual information* are heterogeneous. In our research, we investigate how to reliably transfer semantics from a structural knowledge base to unstructured semantic representations.

Research Question 2 (RQ2): *how to reliably transfer semantics from multiple source domains to a low-resource target domain?*

Many natural language processing tasks are performed in low-resource domains, without enough corpus to learn reliable semantic representations. Inspired by transfer learning methods [19], several works have shown that incorporating multiple word representations (denoted as source embeddings) learned from large-scale corpora can improve the quality of semantic representations (denoted as target embeddings) in the task domain. For example, Tsuboi [21] showed that using Word2Vec [8] and GloVe [7] embeddings together could improve the tagging accuracy. Thus, it is feasible to combine the strengths of multiple source embeddings and yield a new representation space with improved overall expression quality. Most existing methods carry out straightforward mathematical operations over the set of source embeddings, such as concatenation [22], averaging [23], or constructing a new common embedding space by capturing complementary information in different source embeddings [24]. However, these methods treat different source embeddings with various qualities equally, and combine source embedding directly rather than adapting to the need of target tasks. Moreover, previous works did not consider the importance of semantic from the task domain which is crucial to downstream tasks. To alleviate these problems, we dive into how to dynamically aggregate source embeddings with the attention schema to learn semantic representations for the task domain with limited resources.

Research Question 3 (RQ3): *how to achieve reliable and low-cost cross-lingual transfer of semantics?*

Most natural language processing methods are designed for high-resource languages, leaving the vast majority of low-resource languages understudied. When these techniques which are designed for high-resource languages are applied to low-resource languages, the main obstacle is the cross-lingual gap, which is due to the fact that the same meaning has different characters in different languages. Recently, researchers designed cross-lingual transfer methods which aim to use the data and models available in a high-resource language (e.g., English) to solve tasks in another, commonly low-resource language. Most existing methods heavily rely on large-scale parallel bilingual resources, such as translator, cross-lingual mapping lexicon [25], or an intermediary language such

as pivoting [26]. However, collecting parallel bilingual resources is expensive and time-consuming. To be more practical, we investigate how to conduct cross-lingual transfer with limited resource reliance. We explore a reliable and low-cost transfer learning method to help tasks defined on low-resource languages.

Research Question 4 (RQ4): *how to adapt semantic representations for specific applications?*

Semantic representations of text mainly encode the general-purpose features. Considering words have task-oriented features in a specific task, adapting semantic representations to specific tasks has the greatest potential in real-world applications. For example, sentiment analysis [27] emphasizes the emotional polarity of words and text classification [17] emphasizes the category attributes of words. Thus, it is necessary to investigate how to adapt semantic representation to the needs of specific tasks. We focus on the application of a semantic similarity measure which computes the similarity of words using their corresponding vector similarity (e.g., cosine similarity). Despite their effectiveness, the quality of the similarity measured by semantic representations (also known as embedding similarity) is under debate. The crux of the discussion is that most word embedding methods rely on statistics, to show how often each word occurs within the local context window of another word, which means they mainly capture the *proximity* property between words [28] whereas in practice, many valuable associative relationships between words could exist across a longer linguistic distance. Thus, we explore how to adapt semantic representations to cover long-distances relationships and refine the similarity measure according to the real-world applications.

This thesis aims to achieve the following objectives, which are expected to answer the aforementioned research questions:

Research Objective 1 (RO1): *To design a structural knowledge modeling method and propose a semantic structure-based semantic representation method.* (aims to answer RQ1)

Generally, knowledge bases store structural knowledge information in a graph form by representing words as *nodes* and the relationships between words as *edges*. To lever-

age structural knowledge, most previously proposed methods simply use relations within word-pairs, e.g., constraining words belonging to one semantic category [29] or constructing a regularizer to model words in particular semantic relations [30]. As such, these works do not fully explore the comprehensive structures in the knowledge bases. We argue that effective knowledge modeling should contain the whole semantic structures within the knowledge base. We design the principle of preserving semantic structures by converging words to their concept on the upper level and diverging words on the same sense level. The basic idea can be intuitively explained as follows: *football* and *basketball* are related to *ball* (denoted as concept convergence), but they also hold different attributes since they are indirectly linked in the graph (denoted as word divergence). Compared with only modeling relations in word-pairs, our method comprehensively models a word's structural features with its directly linked and indirectly linked words in the knowledge base, which is more stable and reliable.

To propose a new knowledge-enhanced semantic representation method, we address the problem of modeling heterogeneous contextual information and knowledge, based on the assumptions of concept convergence and word divergence. We propose a semantic structure-based word embedding method called SENSE and design a joint learning framework to model contextual information and knowledge. Our method departs from previous work in that it explores the global structural information of words in the use of a knowledge base, not the local relations that exist between two words. We use real-world datasets to evaluate its efficacy.

Research Objective 2 (RO2): *To propose a new meta-embedding method to dynamically leverage semantics from multiple source domains.* (aims to answer RQ2)

To learn semantic representations for a low-resource target domain, we probably have multiple source embeddings trained from different domains. It has been observed that multiple source embeddings vary significantly in the quality and characteristics of the semantics according to the type of training corpus, how the learning architecture is designed, and whether or not external knowledge is considered [31, 32]. Thus, we design a dynamical meta-embedding method to combine these source embeddings for the task

domain.

Moreover, in natural languages, the distributions of salient and domain-specific words often determine the meaning of a document [33], but they rarely appear in other domains. As such, source domains often lack sufficient information for training high-quality embeddings for specific words. It is worth noting that using contextual information alone does not capture the relationships of domain-specific words well, especially in low-resource scenarios. Thus, we model the relationship between salient and specific words in the task domain, and maintain these task-specific features in the learned representation space.

Research Objective 3 (RO3): *To propose a new method to bridge the cross-lingual semantic gap with limited bilingual resource reliance.* (aims to answer RQ3)

Existing methods address cross-lingual semantic gap using two types of methods. Lexicon-based methods [34] leverage translators or bilingual mapping lexicons, which are simple but consume a large amount of resources. Semantic-based methods [35] learn the common semantic representation space of multiple languages, which only rely on a small bilingual dictionary but they perform poorly and are not efficient in representing complex semantics. To propose a new method for the cross-lingual transfer of semantics, we address the key problems of heavy resource dependencies and poor semantic representation. We design a pivot-based method to learn an intermediary set which is semantically related to a low-resource language and lexically related to a high-resource language. Then, we adapt models of the high-resource languages to the low-resource languages with the help of the intermediary set and we evaluate the effectiveness of the proposed method on the cross-lingual entity linking task.

Research Objective 4 (RO4): *To propose a new method to adapt semantic representations according to the task-oriented features.* (aims to answer RQ4)

Existing semantic representation methods learn the general characteristics of text, while specific tasks require task-oriented properties. Thus, to move towards realistic tasks, we extend semantic representations to task-oriented methods. We focus on refining the semantic similarity measure for top-K words selection. In general, the similarity

of text is computed using its corresponding embedding similarity. However, these embeddings are trained only considering the local proximity properties of two words in a corpus. To mitigate this issue, we use association rules to measure word similarity at a global level and propose a fuzzy similarity measure which jointly encodes the local and global similarities. After formalizing the task-oriented semantic similarity measure, we use the query expansion task as the test-bed and evaluate the efficacy of our method on real-world datasets.

1.4 Research Innovation and Contributions

This thesis aims to enhance modeling transferable semantics by addressing the key problems faced by existing semantic representation methods. The main contributions of this study are summarized as follows:

1.4.1 Research Innovation

Innovation 1. This study proposes a novel knowledge-enhanced word representation method which models the whole semantic structures within the knowledge base. Different from previous methods which only model local relations that exist between two words, our method explores global structural information of words in the usage of knowledge base. It is able to transfer stable and reliable knowledge from a knowledge base to the semantic representation space.

Innovation 2. This study provides a new method to dynamically transfer knowledge from multiple source embeddings to produce more expressively powerful semantic representations for a low-resource domain. To handle the challenges of domain shifts and quality discrepancy, the proposed method dynamically aggregates multiple source embeddings to a single meta-embedding space by a differentiable attention module, rather than using them equally. It is more suitable for transferring true required semantics from multiple source domains to a low-resource domain.

Innovation 3. This study designs an effective method to bridge the cross-lingual semantic gap with limited parallel resources. To transfer the models trained on high-resource languages to low-resource languages, the proposed method generates a pivot set which is lexically related to high-resource languages and semantically related to low-resource languages. It is a new and novel solution to transfer knowledge from high-resource language to low-resource language.

Innovation 4. This study proposes a novel fuzzy similarity measure by refining general semantic representations with task-oriented information. In addition to using similarity measure driven from word embeddings, the proposed method also extracts global relatedness information from the task-specific corpus. It designs a fuzzy word similarity measure that relies on a fuzzy logic system to combine complementary but heterogeneous global and local information. It is an efficient way to transform task-oriented characteristics into semantic representations that can improve the performance of downstream applications.

1.4.2 Research Contributions

Contribution 1. A semantic structure-based word representation method, called SENSE, is proposed to jointly leverage knowledge and text to encode word semantics.

1) This study designs the principle of preserving semantic structures by converging words to their concept on the upper level (denoted as concept convergence) and diverging words on the same sense level (denoted as word divergence). We show that this principle is effective and easy to implement in the word embedding training process.

2) This study designs an approach for learning word embedding that considers relatively stable and reliable semantic structures within knowledge bases. We evaluate the proposed method on intrinsic and extrinsic tasks, showing its effectiveness in transferring semantics from knowledge bases to the vector space.

Contribution 2. A dynamic meta-embedding method is proposed to leverage background semantics from multiple source embeddings.

1) This study develops an unsupervised, end-to-end trainable framework that addresses domain-specific meta-embedding by comprehensively exploring the task domain and various source domains. Instead of mining semantics from a single domain, this method leverages knowledge from multiple domains, to generate high-quality and accurate word representations.

2) This study dynamically leverages different source embeddings with an unsupervised attention mechanism. Unlike previous methods, this method provides an effective solution to dynamically combine multiple source embeddings with the attention schema, rather than using them equally.

3) This study explores latent semantics in the task domain in the meta-embedding learning process. Specifically, it applies a graph convolution network (GCN) over a domain-specific graph, which efficiently represents the significance of domain-specific words and their global correlations. As such, this method alleviates the problem of inadequate training on domain-specific words.

Contribution 3. A pivot-based method is proposed to address the cross-lingual semantic gap to transfer the models trained on the source languages to the target languages.

1) This study develops a pivot-based method which bridges the cross-lingual gap with an intermediary set. This set contains source language words which are semantically related to the target language and lexically related to the source language. It is helpful to apply models trained on the source language to the target language.

2) This study leverages the pivot-based method on cross-lingual entity linking. It emphasizes the importance of leveraging pivots to bridge the cross-lingual and mention-entity gaps. Moreover, it inherits the advantages of semantic-based and lexicon-based approaches while avoiding their limitations.

Contribution 4. A fuzzy similarity measure is proposed to address the lack of global relatedness when employing general semantic representations on the query expansion

task.

1) This study proposes an efficient strategy to extract globally related words by encoding complementary global information into traditional local similarity measures derived from word embedding.

2) This fuzzy word similarity measure relies on a fuzzy logic system to combine complementary but heterogeneous global and local information from a corpus. Several new fuzzy rules are designed to infer word similarity, based on both local and global measures as inputs.

1.5 Research Significance

The theoretical and practical significance of this thesis is summarized as follows:

Theoretical significance: This thesis investigates transferable semantics to improve word representations. The key idea of this thesis is to model transferable semantics in three aspects: 1) discovering high-quality semantic knowledge from the source domain (e.g., knowledge bases and pre-trained source embeddings); 2) modeling semantic knowledge effectively; and 3) adapting semantic knowledge to the target domain (e.g., low-resource domains and low-resource languages). These theoretical results have the greatest potential to guide future researchers to develop more powerful semantic representation methods.

Researchers can follow the proposal of the semantic structure-based representation method to convert a semantic graph to the representation learning process, which will enable the semantic representation to be applied to address more knowledge-related problems. Furthermore, using the assumptions of concept convergence and word divergence that are used to model semantic structure, researchers can also improve the encoding properties of semantic graphs.

To leverage multiple source embeddings, this thesis designs a dynamic meta-embedding approach, which selectively combines different source embeddings and emphasizes the characters of the target domain. Based on the theoretical results presented in this thesis,

in the future, researchers can develop more methods to leverage multiple pre-trained language models to support downstream tasks.

To bridge the cross-lingual semantic gap, this thesis generates pivots to connect different languages semantically and lexically, which is helpful to apply the testing data of target languages on models trained on source languages. Based on this finding, in the future, researchers can develop more methods based on aligned multilingual embeddings to address the cross-lingual semantic gap with limited parallel data.

Researchers can follow the proposal of the fuzzy semantic similarity measure to adapt generic semantic representations to downstream tasks, which refines generic semantic representations to capture task-specific features. Furthermore, researchers can develop more task-specific adaption methods to leverage the semantic representations to help the task defined on a specific domain.

Practical significance: The findings of this research will benefit society given the increasing demand for text-related applications in modern life. First, this study presents word representation methods for *low-resource* domains by transferring semantics from knowledge bases, multiple source domains, and high-resource languages. Thus, it is benefit to solve real-world applications with limited resources. Second, this study reveals the importance of *task-specific* and *domain-specific* information in learning word representations. Moreover, this study also provides a method to adapt generic semantic representations to a specific task considering the task-oriented features. Last, all these methods are validated by real-world datasets, which means practitioners can directly use the proposed methods to solve real-world problems. These findings help resolve real-world natural language processing tasks. There is potential for many other applications to benefit from this study, such as knowledge representation and semantic computing.

1.6 Thesis Structure

The structure of the thesis is shown in Fig. 11 and the chapters are organized as follows:

- CHAPTER 2 presents the literature on semantic representation, thereby revealing the limitations of the current research.
- CHAPTER 3 presents a novel semantic structure-based word embedding method, and introduces concept convergence and word divergence to reveal semantic structures in the word embedding learning process. This chapter addresses RQ1 to achieve RO1 when transferring semantics from structured knowledge bases.
- CHAPTER 4 presents a novel dynamic meta-embedding method which jointly models background knowledge from the source embeddings and domain-specific knowledge from the task domain. This chapter addresses RQ2 to achieve RO2 when transferring semantics from multiple domains to a low-resource domain.
- CHAPTER 5 presents a pivot-based method to bridge the cross-lingual semantic gap, which is helpful to transfer models trained on source languages to the target languages. This chapter addresses RQ3 to achieve RO3 when transferring semantics across different languages.
- CHAPTER 6 presents a fuzzy similarity measure to address the lack of global relatedness when employing general semantic representations on the query expansion task. This chapter addresses RQ4 to achieve RO4 when refining semantic representations for real-world applications.
- CHAPTER 7 summarizes the findings of this thesis and points to directions for future work.

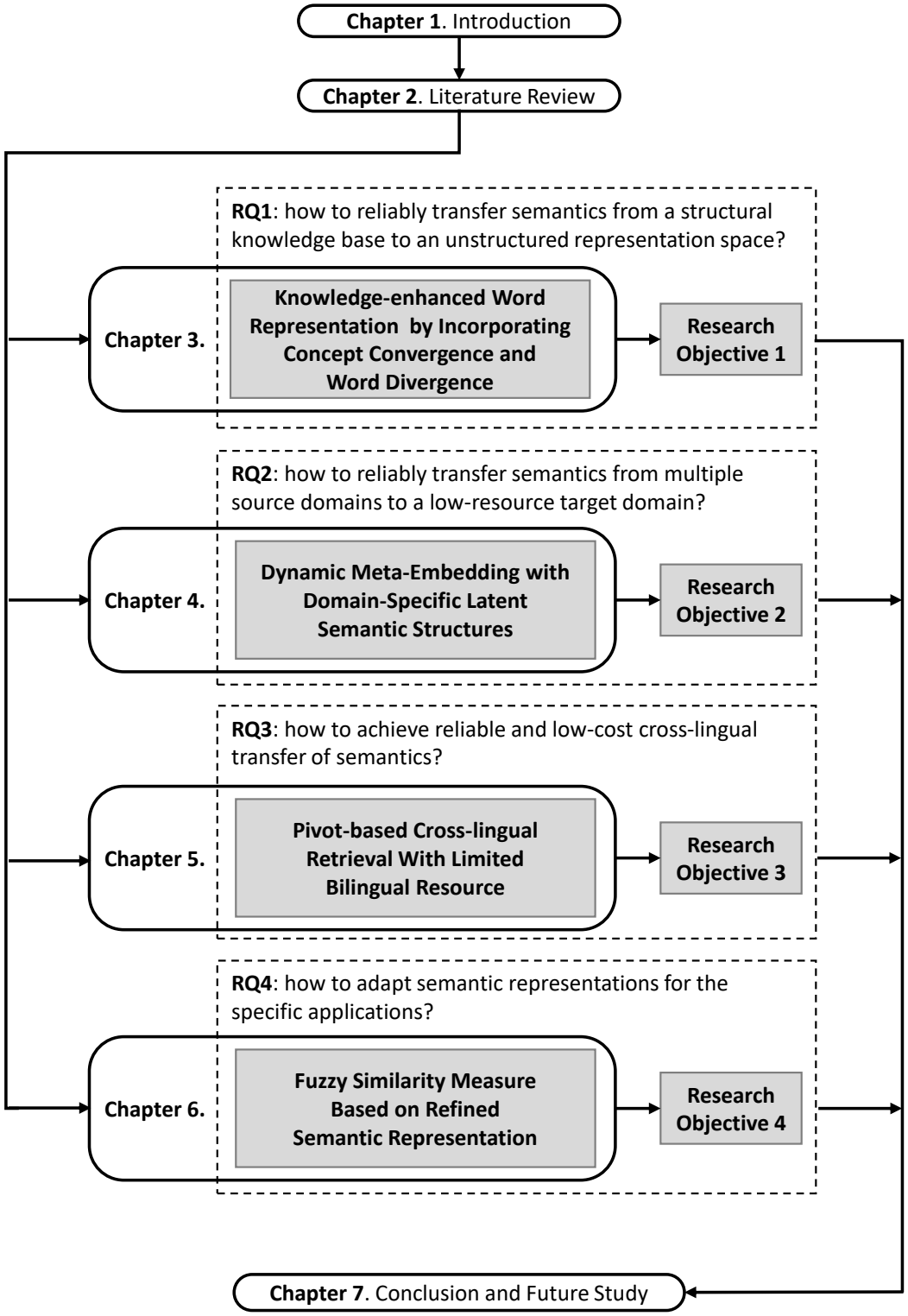


Figure 11: Thesis structure

CHAPTER 2

LITERATURE REVIEW

In this chapter, we review recent works related to this thesis, including three aspects: semantic representations, transfer learning, and natural language processing applications.

2.1 Semantic Representation

2.1.1 Distributed Representation

Representing words using fixed-length vectors is an essential step in text processing tasks. In the early stage, one-hot representations have been widely used for its simplicity and efficiency. However, this traditional representation method suffers from data sparsity, the curse of dimensions and lexical gap, which make NLP and IR tasks difficult to use. Distributed word representation, also known as word embedding, is then introduced to solve these problems. In distributed representation methods, words are represented as dense, low-dimensional, real-valued vectors, and each dimension represents latent semantic and syntactic features of words. As an improvement of traditional one-hot word representation method, word embedding overcomes the data sparsity, high dimensional,

and lexical gap problems by capturing both word semantics and syntactics with dense vectors.

The underlying idea of distributed representation is that *"words with similar context tend to have similar meaning"* [36]. There are mainly two approaches: 1) *count-based* methods which model word co-occurrence statistics to the vector space; and 2) prediction-based methods which predict a word from its context. The early representative count-based methods are the Latent Semantic Analysis (LSA) [6] and Latent Dirichlet Allocation (LDA) [37], which learn low-dimension vectors using singular value decomposition. Recently, Pennington et al. [7] proposed the Global Vectors model (GloVe), which is an unsupervised learning algorithm to learn word representations based on aggregated global word-word co-occurrence statistics from a corpus. The prediction-based methods are generally implemented with neural network language modeling. Bengio et al. [38] proposed a Neural Network Language Model (NNLM) by designing a multiple-layer neural network to train the language model, and word vectors are parameters of the network. Mnih and Hinton [39] improved the NNLM model by removing the non-linear activation function. Recently, Mikolov et al. [8] proposed the Word2Vec method, which is the most effective word representation method. There are two models, i.e., CBOW predicts the central word by its context, and Skip-gram model predicts the context using the central word. Following this trend, distributed word embeddings have been widely studied in NLP area [40–42].

Beyond context information, many researcher try to improve word embeddings considering other information of words, such as topic information [15, 43], sentiment information [18, 44], task-specific information [17, 45], and morphological knowledge. In addition, there are also works aimed at incorporating explicit knowledge into embedded words [46]. For example, Bollegala et al. [47] incorporated the *IsA* relation of words to the embedding space. Faruqui et al. [48] considered semantic relations from WordNet [13] and Paraphrase Database [49], and they proposed a retrofitting method to encourage linked words have similar representations in the embedding space.

2.1.2 Contextual Representation

Despite effectiveness, distributed representations encode each word using a static vector regardless of its context. In general, words have different meanings under different context, thus it is not reasonable to use a fixed vector to represent words. Moreover, word-level representations are limited in their ability to express long text (such as sentence and document). To solve these problems, researchers designed pre-trained language models (PTMs) which are contextual representation based on deep models (i.e., Transformer [50]). Several representative PTMs are detailed in this section.

ELMo [9] (Embeddings from Language Models) is groundbreaking work which first learns contextual word embeddings based on the entire sentence. ELMo contains a deep bidirectional language model, which is pre-trained on a large text corpus. At the time of its release, ELMo achieved state-of-the-art performance in the various reasoning benchmarks of the time, including question answering, co-reference resolution, and text entailment. These achievements suggested that deep internals of the pre-trained network are indeed useful for text-related applications, and inspired the development of subsequent contextual representation methods.

GPT [51] (Generative Pre-trained Transformer) introduced Transformer [50] to learn deep language model. Compared with just using pre-trained language as input features for downstream tasks (e.g., ELMo), GPT demonstrated that large gains can be achieved by generative pre-training of a language model on unlabeled corpora and then fined-tuned using supervised and task-specific data. To improve the generalization ability, GPT-2.0 [52] and GPT-3.0 [53] with significantly more parameters are released, and they achieved performance on various benchmarks in the zero-shot, one-shot, and few-shot settings.

BERT [10] (Bidirectional Encoder Representations from Transformers) is a widely used pre-trained language model. Different from classic language model which predicts the next token given context, BERT is implemented with the *masked language model* task which empowers BERT with deep bidirectional representations ability by jointly conditioning on both left and right context. Moreover, BERT also added the *next sen-*

tence prediction task to understand the relationships between sentences. Further, the researchers developed several variants that further enhanced the model performance, such as RoBERTa [11] and ALBERT [54].

Pre-trained language models mainly learn universal language representation from large-scale plain corpora but rarely consider knowledge. Recently, researchers have develop several pre-trained language models to learn knowledge-enriched language models. For example, ERNIE [55] (Enhanced Language Representation with Informative Entities) integrates entity embeddings pre-trained on a knowledge graph with corresponding entity mentions in the text, to capture lexical, syntactic, and knowledge information simultaneously. KnowBERT [56] trains BERT jointly with an entity linking model to incorporate entity representation in an end-to-end fashion. LIBERT [57] (Linguistic Informed Multi-Task BERT) incorporates linguistic knowledge via an additional linguistic constraint task. SenseBERT [58] introduces lexical semantic information to the pre-train language model. It predicts not only the masked words but also their WordNet supersenses. K-BERT [59] (Knowledge-enabled Language Representation Model) explicitly injects related triples extracted from knowledge base into the sentence to obtain an extended tree-form input for BERT. It can enable language representation with knowledge graphs, achieving the capability of commonsense or domain knowledge. SentiLR [60] is proposed to capture not only the context dependency but also the linguistic knowledge from SentiWordNet, by designing label-aware masked language model to enable the pre-trained model to utilize the knowledge in sentiment analysis tasks. KEPLER [61] (Knowledge Embedding and Pre-trained LanguagE Representation) is proposed to better integrate factual knowledge into pre-trained language models but also produce effective text-enhanced knowledge embedding with the strong language models.

There are three main advantages of pre-training language models [62]: 1) learning high-quality universal language representations; 2) providing a better model initialization instead of training a new model from scratch; 3) avoiding over-fitting problem for tasks with small data.

In this thesis, we mainly focus on the improvement on distributed representations

considering that PTMs are resource-intensive and relies on industrial-grade computing power (e.g., GPUs). We designed methods to explore and leverage transferable semantics from knowledge bases and multiple source domains, bridge cross-lingual semantic gaps, and adapt general representations to specific tasks. These proposed methods and findings are also informative to improve the pre-trained language models.

2.2 Transfer Learning

This thesis aims to improve semantic representations by leverage knowledge beyond raw text. The underlying theory is *transfer learning* which is a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. We first briefly introduced the related work of transfer learning. Then the representative work of several semantic representation methods based on transfer learning is introduced in detail.

Transfer learning aims to utilize previously-acquired knowledge to solve new but similar problems [19]. A variety of real-world applications have been benefited from the recent advances of domain adaptation (DA), such as sentiment analysis [63], cross-language text classification [64, 65]. Previous research has expended effort on constructing domain specific distributed representations of words using transfer learning methods. Wang et al. [66] proposed a unified hierarchical merging approach built upon the graph-embedding framework, which is able to merge visual words for any scenario, where a preferred structure and an undesired structure are defined, and, therefore, can effectively attend to all kinds of requirements for the word-merging process. Zheng et al. [67] presented a neural network architecture as well as a context-specific model that can learn multi-prototype word/character representations, which are capable of capturing word's or character's syntactic and semantic information, particularly their polygamous variants. However, there are limited studies on transferring semantic knowledge across heterogeneous domains, such as transfer the graph-organized relations in the knowledge bases to the unlabeled corpus.

There are also some works considered incorporating external knowledge bases constructed by human experts into word embeddings. Several studies use combined methods to fit pre-trained word embeddings with the given external resource, making no assumptions about how the input embeddings were constructed. For example, the *Retrofit* method [48] refines word embeddings using relational information, which encourages linked words have similar vector representations in the embedding space. Goikoetxea et al. [68] learned word representations from text and WordNet independently, and then explored both simple and sophisticated methods to combine them, showing that a simple concatenation of independently learned embeddings outperforms more complex combination techniques in word similarity and relatedness datasets. In contrast to the combined methods, several studies have jointly leveraged semantic lexicons and corpus-based methods. The RCM method [29] is a relation constrained model which introduces a training objective that incorporates both a neural language model objective and a semantic knowledge objective. In the RCM method, the knowledge base functions as word similarity information to improve the performance of word embedding. Xu et al. [69] leveraged both relational and categorical knowledge to produce word representation (RC-NET), combining this with the Skip-gram method. Bollegala et al. [47] proposed a method that considers semantic relations in which they co-occur to learn word representations. Bollegala et al. [30] also proposed a joint word representation learning method that simultaneously predicts the co-occurrences of two words in a sentence, subject to the relational constraints given by a semantic lexicon. Although these studies consider the semantic information from an external knowledge base in the learning process, they do not leverage high-quality semantic structures to improve word embeddings.

There are still several unsolved challenges in improving semantic representations with knowledge beyond raw text. For example, how to capture heterogeneous knowledge and use it reliably and efficiently for semantic representation (Chapter 3); how to leverage knowledge from multiple sources (Chapter 4); how to transfer knowledge in a cross-lingual scenario (Chapter 5). In this thesis, we provide effective solutions and design new methods to address these challenges.

2.3 NLP Applications

Word representations have been extensively used for various natural language processing tasks. Recently, substantial work has shown that task-oriented features can further improve word representations, which are beneficial for downstream NLP tasks.

Grover and Mitra [70] proposed a novel model which can be used to align the sentences of two different languages using neural architectures. Aydin et al. [71] proposed automatic query generation method using word embeddings for retrieving passages describing experimental methods. Yao et al. [72] incorporated word embeddings obtained from a large number of domains into topic modeling. By combining Latent Dirichlet Allocation, a widely used topic model with Skip-gram, a well-known framework for learning word vectors, the proposed method could improve the semantic coherence significantly. Tao et al. [73] utilized word embeddings and tackled the task of extracting prescriptions from discharge summaries in two extraction steps, both of which are treated as sequence labeling problems.

Specially, several researcher pay attention into task independent fine tuning for word embeddings. Yang and Mao [74] proposed a task-independent fine-tuning framework, where the task-independent fine tuning is to integrate multiple word embeddings and lexical semantic resources to fine tune a target word embedding. In query expansion tasks, the top-k most similar terms in the neighborhood of a given query are generally selected as the related terms. The embedding similarities are regarded as weights and incorporated into the retrieval model. A series research [75–78] has shown that word embeddings can be properly integrated in query expansion and further improve the information retrieval effectiveness. However, it is still a challenging problem that terms with high embedding similarities may not fit the needs of query expansion, and then damage the retrieval performance [79]. Some recent studies aim to better adapt word embedding-based query expansion methods to the needs of information retrieval. Rekabsaz et al. [80] explore the embedding similarity space and suggest a general threshold to filter the most effective related terms. Zamani and Croft [28] show that the relatedness in word embeddings is not match the goal of query expansion. They develop

unsupervised relevance-based word embedding models that learn word embeddings based on query-document relevance information.

To conclude, when generic word representations are used in downstream tasks, it is important to use task-oriented characteristics to fine-tune semantic representations. In this thesis, we also emphasis on the task-orient features in the learning process of word representations: 1) we explore how to maintain domain-specific semantic features in the semantic transfer process (Chapter 4); and 2) we explore how to adapt generic word representations for a specific task (e.g., top-k selection in Chapter 6).

CHAPTER 3

KNOWLEDGE-ENHANCED WORD REPRESENTATION BY INCORPORATING CONCEPT CONVERGENCE AND WORD DIVERGENCE

3.1 Introduction

Understanding and representing the sense of text is a fundamental task in both information retrieval (IR) and natural language processing (NLP). Previous research has expended great effort on constructing distributed representations of words (also known as word embedding) as the atomic components of text by embedding the semantic and syntactic properties of the surface text into low-dimensional dense vectors. Trained word embeddings have achieved overwhelming success in various real-world applications, e.g., document retrieval [81–83], text classification [84], question answering [85], and sentiment classification [86].

Most of the research directs attention entirely towards learning word representation methods from a large unlabeled corpus, such as *prediction-based* methods [8, 20, 87, 88] which learn word representation by predicting the co-occurrence of words in the given

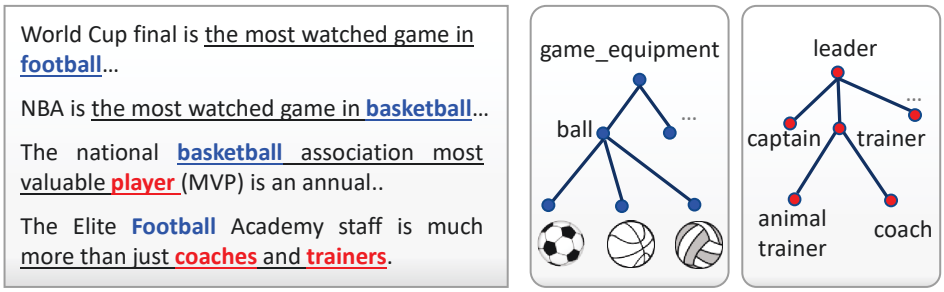


Figure 31: We mark two types of words in the sentences with blue color and red color, respectively. The underlined words are their context in the corpus. The graphs in the right are their semantic structures generated from WordNet.

context, and *counting-based* methods [7] which learn word representations through global matrix factorization based on a count of co-occurring words. These corpus-based methods mainly consider a word's co-occurrence information and, therefore, generally learn similar embeddings for words with similar contexts.

In the past few years, some efforts have focused on learning word representation beyond the corpus, and considered external knowledge bases constructed by human experts, such as semantic lexicons and concept graphs [30, 47, 68, 89, 90]. Most previously proposed methods simply use relations within word-pairs, e.g., constraining words belonging to one semantic category [29], or constructing a regularizer to model words in particular semantic relations [30]. As such, this work did not fully explore the comprehensive structures in the knowledge bases (KBs).

In this chapter, we argue that effective word embeddings should contain the semantic structures within the knowledge base. We illustrate how the semantic structures can be a complementary source for word embeddings in Fig.31. As shown in the sentences, *football*, *basketball*, *trainer*, *coach* usually share similar context, and tend to have similar representations in the corpus-based methods. While the semantic structures in the right side clearly define these words with different semantic granularities and abstractions, i.e., these four words are located in two different subgraphs, showing that they belong to different concepts; *football* and *basketball* are not directly linked in the subgraph,

showing that they hold different attributes. On the other hand, compared with relations in word-pairs, comprehensively modeling a word's structural features with its directly linked and indirectly linked words in the KBs could be more stable and reliable [91, 92].

To this end, we propose a **se**ma**n**tic **s**tructure-based word **e**mbedding method called SENSE. Moreover, we introduce *concept convergence* and *word divergence* to implement semantic structure modeling in the word embedding learning process. The basic idea can be intuitively explained as *football* and *basketball* are related to *ball* (concept convergence), but they also hold different attributes since they are indirectly linked in the graph (word divergence). This work departs from previous works in that it explores global structural information of words in the usage of knowledge base, not the local relations that exist between two words. We evaluate our word embedding method using extensive intrinsic and extrinsic evaluations. The experimental results show that modeling semantic structures in the knowledge base by incorporating concept convergence and word divergence makes embeddings significantly more powerful, and results in consistent performance improvement across real-world applications.

3.2 Background

The last few years have seen the development of distributed word representation learning methods purely based on the co-occurrence information in a corpus [7, 8, 38, 39, 87, 93]. Some recent studies throw light on the semantic knowledge stored in the KBs, showing that the KBs can potentially assist the word embedding learning process.

Several studies use combined methods to fit pre-trained word embeddings with the given external resource, making no assumptions about how the input embeddings were constructed. For example, the *Retrofit* method [48] refines word representations using relational information from semantic lexicons. The method encourages linked words to have similar vector representations which are then embedded in a semantic network that consists of linked word senses in a continuous-vector word space. Goikoetxea et al. [68] learned word representations from text and WordNet independently, and

then explored both simple and sophisticated methods to combine them, showing that a simple concatenation of independently learned embeddings outperforms more complex combination techniques in word similarity and relatedness datasets.

In contrast to the combined methods, several studies have jointly leveraged semantic lexicons and corpus-based methods. The RCM method [29] is a relation constrained model which introduces a training objective that incorporates both a neural language model objective and a semantic knowledge objective. In the RCM method, the knowledge base functions as word similarity information to improve the performance of word embedding. Xu et al. [69] leveraged both relational and categorical knowledge to produce word representation (RC-NET), combining this with the Skip-gram method. Liu et al. [90] represents semantic knowledge as a number of ordinal similarity inequalities of related word pairs to learn semantic word embedding (SWE). Bollegala et al. [94] proposed a method that considers semantic relations in which they co-occur to learn word representations. Although these studies consider the semantic information from an external knowledge base in the learning process, they do not leverage high-quality semantic structures to improve word embeddings.

Our work can be categorized as a joint learning method that incorporates both co-occurrence information and semantic structures. In contrast to the aforementioned research, we leverage the semantic structure information in the KBs. In our method, we construct multi-level structures from the knowledge base to express semantic granularity and abstraction. Moreover, we design principles of concept convergence and word divergence to implement semantic structures into the word embedding learning process.

3.3 Semantic Structure-based Word Embedding

Given a corpus $\mathscr{C}$ and a knowledge base $\mathscr{G}$ as input, the SENSE method learns a d dimensional vector $\mathbf{x}_w \in \mathbb{R}^d$ for each word w in the corpus. Any KB that captures the relationships between words in a hierarchically-organized manner could be used to generate semantic structures, such as WordNet [13], Freebase [95], and PPDB [96]. We

use WordNet to describe the method and conduct the experiments.

The KB is defined as a directed graph $\mathscr{G} = (V, E)$, where the set of vertices V denotes words, and the set of edges E denotes the semantic relations between the pairs of vertices. Intuitively, a vertex's structure information in a directed graph can be covered by exploiting its parent vertices, brother vertices, and child vertices. Fig. 32 visualizes the structures of the word *dog* in WordNet. Our ideas for modeling the structures were inspired by the observations in the nature language.

First, words directly linked in semantic structures share the same attributes. For example, *canine* is the parent of *dog* and *wolf*, and *canine* can be regarded as a concept that represents the common attributes of a *dog* and a *wolf*. These directly linked words tend to converge, and the child words tend to be close to the parent word. Thus, we assume that:

Assumption 1. *Concept convergence: The upper level is regarded as the concept of its lower level. The center of all words in the lower level tends to converge to their upper-level word.*

Second, brother words in semantic structures are indirectly linked and are located in the same level. They tend to be diverged, giving the areas of different words a distinct positioning for different attributes. For example, *wolf* and *dog* are close to *canine* as they share the same attribute, but they should be separated from each other since they also hold significantly different attributes. Thus, we assume that:

Assumption 2. *Word divergence: Words in the same level hold distinctive attributes, and they tend to be diverged.*

A variety of corpus-based methods have been proposed to learn word representations by optimizing the prediction ability between words and contexts. We follow the Word2Vec method, which uses extremely computationally efficient log-linear models to produce high-quality word embeddings. The Word2Vec method applies a sliding window moving on the corpus, and the central word is the target word and the others are context words. There are two models: the CBOW model uses the average/sum of context words as input

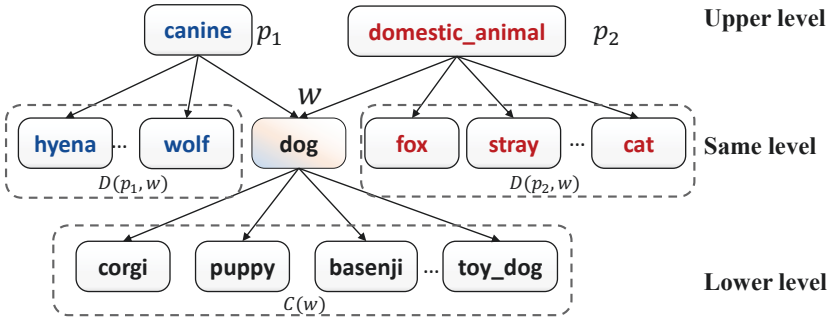


Figure 32: An example of the three-level semantic structures of the word *dog* in WordNet.

to predict the target; the Skip-gram model uses the target word as input to predict each context word. To simplify, we represent the objective of each prediction as

$$\mathscr{L}_{context} = Pr(w|\mathbf{c}) = \frac{\exp(\mathbf{x}_w \cdot \mathbf{c})}{\sum_{w' \in \mathcal{V}} \exp(\mathbf{x}_{w'} \cdot \mathbf{c})}, \tag{3.1}$$

where w is the predicted word, $\mathbf{c}$ is the low-dimensional real-value vector of input word/words, $\mathbf{x}_{w'} \in \mathbb{R}^d$ is the vector representation of the word w' in the vocabulary $\mathcal{V}$.

The objective of the SENSE method is to train word representations that are not only good at predicting its context words, but are also good at modeling concept convergence and word divergence. Let w represent the predicted word in each prediction task. We detail how to represent structural information of word w in $\mathscr{G}$.

Specially, we define $\mathscr{G}$ using WordNet, where words are grouped into sets of cognitive synonyms (denoted as synsets), and synsets are interlinked by hyponym-hypernym relations (i.e., general terms and specific kinds). We observe that WordNet is a complex hierarchical graph of synsets: (1) each word points to at least one synset. Hence, there is a many-to-many relationship between synsets and words; (2) the synset would have more than one parent in WordNet. In our method, we model the semantic structures on the granularity of synsets. Formally, given a word, we denote its synset collection as $S = \{w^1, \ldots, w^k\}$, where $w^i (1 \le i \le k)$ represents one synset of the word, denoted as w for brevity. Then for each synset of word w, we exploit the following three-level features that capture varying granularity semantic structures:

- Let $P(w) = \{p_1, \ldots, p_{|P|}\}$ represent the collection of words on the upper level of word w, where $p_i \in V$, and the edge $< p_i, w >$ exists in E.

- Words on the same level of w are divided into $|P(w)|$ subsets regarding different parent words. Each subset is denoted as $D(p_i, w) = \{u_1, \ldots, u_{|D|}\}$, where $u \in V$, and the edge $< p_i, u >$ exists in E.
- Words on the lower level are specific terms of w, denoted as $C(w) = \{v_1, \ldots v_{|C|}\}$, where $v \in V$, and the edge $< w, v >$ exists in E.

Based on the concept convergence assumption described above, we assume that w should be close to the center of words on the lower level of w (i.e., words in $C(w)$). The training objective is defined to maximize the following function:

$$\mathscr{L}_c = \sum_{S(w)} cos(\mathbf{x}_w, \frac{1}{|C|} \sum_{v \in C(w)} \mathbf{x}_v), \tag{3.2}$$

where $|C|$ is the size of collection $C(w)$. Here $cos(\cdot, \cdot)$ represents the similarity measure function. Following the recommendations in prior work on word similarity measurement, we apply the cosine similarity of a pair of words w_a, w_b by computing

$$cos(\mathbf{x}_{w_a}, \mathbf{x}_{w_b}) = \frac{\mathbf{x}_{w_a}^T \cdot \mathbf{x}_{w_b}}{|\mathbf{x}_{w_a}| \cdot |\mathbf{x}_{w_b}|}. \tag{3.3}$$

The word divergence assumption is defined as enlarging the distance between w and words in the same level with w (i.e., words in $D(\cdot, w)$), and the training objective is to minimize the following function:

$$\mathscr{L}_d = \sum_{S(w)} \sum_{p_i \in P(w)} \sum_{u \in D(p_i, w)} cos(\mathbf{x}_w, \mathbf{x}_u), \tag{3.4}$$

where $P(w)$ is the collection of w's upper level. Because some words have many brother words in KBs, we randomly select several words in the training step. We find that selecting five words is an acceptable trade-off between the method's performance and training speed.

As mentioned before, we integrate the context information and the semantic structure information into a unified framework. Then the new optimization objective is

$$\mathscr{L} = \max_{\Theta}(\mathscr{L}_{context} + \alpha \mathscr{L}_c - \beta \mathscr{L}_d), \tag{3.5}$$

where Θ is a set of all the parameters related to this task, α and β are hyper-parameters, which control the contributions of semantic structures in word embedding learning.

Using the optimization method in [8], we apply negative sampling to solve the context prediction function. If the predicted word w has semantic structures in the KB, the corresponding optimization process for modeling the semantic structures will be activated. The optimization is as follows:

$$
\begin{aligned}
\frac{\partial \mathscr{L}}{\partial \mathbf{x}_w} &= \alpha \frac{\partial \mathscr{L}_c}{\partial \mathbf{x}_w} - \beta \frac{\partial \mathscr{L}_d}{\partial \mathbf{x}_w} = \sum_{S(w)} (\alpha \frac{\partial cos(\mathbf{x}_w, \overline{\mathbf{x}})}{\partial \mathbf{x}_w} \\
&\quad - \beta \sum_{p_i \in P(w)} \sum_{u \in D_{(p_i,w)}} \frac{\partial cos(\mathbf{x}_w, \mathbf{x}_u)}{\partial \mathbf{x}_w}), \\
\frac{\partial \mathscr{L}}{\partial \mathbf{x}_v} &= \alpha \frac{\partial \mathscr{L}_c}{\partial \mathbf{x}_v} = \sum_{S(w)} \alpha \frac{\partial cos(\mathbf{x}_w, \overline{\mathbf{x}})}{\partial \overline{\mathbf{x}}}, \\
\frac{\partial \mathscr{L}}{\partial \mathbf{x}_u} &= -\beta \frac{\partial \mathscr{L}_d}{\partial \mathbf{x}_u} = \sum_{S(w)} \sum_{p_i \in P(w)} \sum_{u \in D(p_i,w)} -\beta \frac{\partial cos(\mathbf{x}_w, \mathbf{x}_u)}{\partial \mathbf{x}_u},
\end{aligned}
\tag{3.6}
$$

where w is the predicted word, u is the word in $D(\cdot, w)$, v is the word in $C(w)$, and $\overline{\mathbf{x}}$ is the average vector of words in $C(w)$. Since we apply the cosine distance to compute the similarity between two words, the optimization can be derived as follows:

$$
\frac{\partial cos(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_i} = -\frac{S_{ij} \cdot \mathbf{x}_i}{|\mathbf{x}_i|^2} + \frac{\mathbf{x}_j}{|\mathbf{x}_i| \cdot |\mathbf{x}_j|},
\tag{3.7}
$$

where $S_{i,j} = \frac{\mathbf{x}_i^T \cdot \mathbf{x}_j}{|\mathbf{x}_i| \cdot |\mathbf{x}_j|}$.

The pseudo code for our word embedding learning method is shown in Algorithm.1. For the efficiency of the algorithm, the training objectives of semantic structure $\mathscr{L}_c$ and $\mathscr{L}_d$ are only activated when the related words have semantic knowledge in KB. Moreover, when there have many related words in KB, we randomly select five words which is an acceptable trade-off between the method's performance and training speed. As such, in our implementation, the optimization process is conducted through stochastic gradient descent (SGD) in a mini-batch mode, with a computational complexity comparable to the optimization process in the Word2Vec method.

Algorithm 1 SENSE method.

Input: WordNet G, Corpus C, dimensionality d of the word embeddings, word vocabulary $\mathcal{V}$

Output: Embeddings $\mathbf{x}_w \in \mathscr{R}^d$ of all words in the vocabulary $\mathcal{V}$.

Initialization: randomly set $\mathbf{x}_w \in \mathscr{R}^d$ for all words $w \in \mathcal{V}$; generate the semantic structures of each word in G; constructing T prediction tasks using a sliding window.
for t = 1, 2, . . . , T **do**
  optimizing $\mathscr{L}_{context}$ using negative sample method introduced in [8]
  if w in G **then**
    use Eq.(3.6) to update $\mathbf{x}_w$, $\mathbf{x}_u$, $\mathbf{x}_v$.
  end if
end for
return $\mathbf{x}_w$ for all words $w \in \mathcal{V}$.

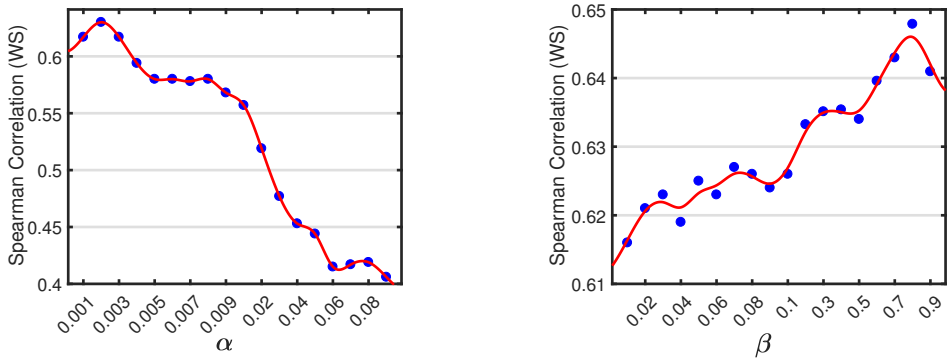


Figure 33: Performance of the SENSE method with varying parameters of α and β.

3.4 Experiments and Results

In this section, we first evaluate the SENSE method's ability to capture semantic and syntactic properties of words. Then, we conduct experiments on the text classification task and the query expansion task, showing that the proposed method boosts performance in real-world applications. The source code of our method is available in the GitHub[1].

3.4.1 Initialization and Parameters

We utilize WordNet (version 3.0) as the KB and use the semantic structure information when words are linked using hypernym-hyponym relation. Since only nouns and verbs

[1] https://github.com/qianliu0708/SENSE

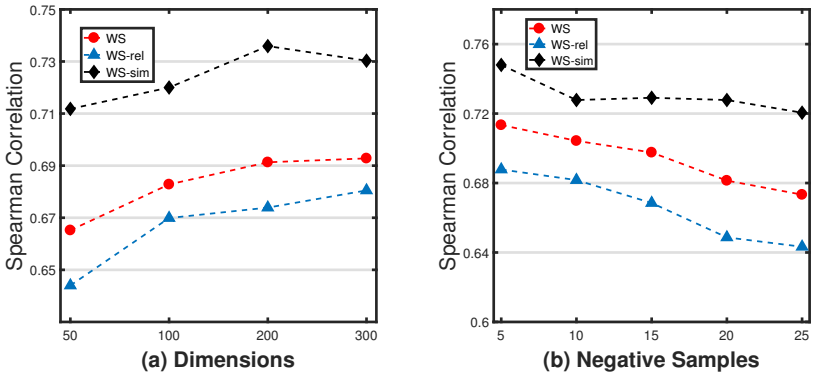


Figure 34: Performance over varying parameters on the WordSim 353 dataset.

hold a hypernym-hyponym relation in WordNet, we extract all the nouns and verbs in WordNet to construct the graph $\mathcal{G}$, resulting in 66,765 nouns with 82,115 synsets and 7,440 verbs with 13,767 synsets.

There are two hyper-parameters in the SENSE method, i.e. α and β in Eq.(3.5), which control the contributions of the semantic structures to the joint learning process. We carefully tune these parameters by fixing one and varying the other. The parameters corresponding to the best word similarity metric value (detailed in next subsection) are used to report the final settings. As shown in Fig.3.4 and Fig.3.4, the SENSE method reaches optimal performance when $\alpha = 0.002$ and $\beta = 0.8$. We follow the optimal settings in this work, with recommended settings of $\alpha \in (0.001, 0.003)$ and $\beta \in (0.7, 0.9)$.

For a fair comparison, all word embeddings adhere to the following settings: the dimensional of vectors is 300, the size of the context window is 5, the number of negative samples is 5, and all KB-enhanced methods are trained using WordNet. Specially, to understand the robustness of our method, we explore the relation between the performance of our method on the word similarity task with varying number of dimensions and negative samples. As shown in Fig.34, we observe that our method is stable when the dimension is set to a value between 100 and 300. The best performance is obtained when the dimension is set to 300. Regarding the size of negative samples, our method obtains optimal results when the number of negative samples is set to 5, and the performance of our method degrades when the number of negative samples is too large.

3.4.2 Word Similarity and Word Analogy

3.4.2.1 Baselines

We compare the SENSE method against two classes of baselines:
(1) **The corpus-based methods** which train word embeddings solely on the corpus. We use the current state-of-the-art methods, including:

- CBOW[2] [8] is a neural network language model which learns word embeddings by maximizing the conditional probability of a target word given the context.
- Skip-gram[3] [8] is a neural network language model which learns word embeddings by maximizing the conditional probability of a context word given the target word.
- GloVe[4] [7] is a state-of-the-art matrix factorization method. It leverages global count information aggregated from the entire corpus as word-word occurrence matrix to learn word embeddings.

(2) **The KB-enhanced methods** which train word embeddings both on the corpus and the KBs. To make a comprehensive comparison, we compare the SENSE method against popular and powerful methods which also use the external KBs, including:

- RCM[5] [29] is a relational constrained word embedding method. It incorporates both the objective of context prediction (following CBOW method and Skip-gram method) and the objective which constrained the relations from the KBs.
- Retrofit[6] [48] is a popular method that refines pre-trained word embeddings using relational information from the KBs.
- Jointreps[7] [30] is a method jointly trained on a word co-occurrence matrix from the corpus (following the GloVe method) and semantic relations from KBs.

[2]http://code.google.com/p/word2vec
[3]http://code.google.com/p/word2vec
[4]http://nlp.stanford.edu/projects/glove/
[5]https://github.com/Gorov/JointRCM
[6]https://github.com/mfaruqui/retrofitting
[7]https://github.com/Bollegala/jointreps

Table 31: Results on the word similarity task and the word analogy task. The word embedding methods are divided into three groups. Bold scores are the best within the groups. Underlined scores are the best overall.

	Word Similarity							Word Analogy		
Methods	**MC**	**MEN**	**RG**	**VERB**	**WS**	**WS-rel**	**WS-sim**	**Sem**	**Syn**	**Tot**
GloVe	0.459	0.506	0.374	0.293	0.509	0.546	0.538	63.5	33.3	56.8
Retrofit-GloVe	0.566	0.526	0.469	0.225	0.539	0.517	0.599	45.3	24.1	42.2
Jointreps	0.394	0.429	0.340	0.308	0.465	0.384	0.534	11.5	6.9	8.8
CBOW	0.641	0.658	0.654	.402	0.638	0.615	0.708	48.2	41.6	48.7
RCM-CBOW	0.492	0.411	0.448	0.247	0.496	0.399	0.569	21.9	11.5	15.1
Retrofit-CBOW	0.677	0.654	0.673	0.365	0.639	0.612	0.711	36.5	38.5	39.1
SENSE-CBOW	0.692	**0.665**	**0.685**	.402	**0.688**	**0.657**	**0.719**	**49.9**	42.2	**49.9**
Skip-gram	0.640	0.676	0.682	0.343	0.631	0.621	0.695	62.4	33.6	56.0
RCM-Skipgram	0.478	0.416	0.418	0.261	0.481	0.393	0.544	21.8	10.9	14.7
Retrofit-Skipgram	0.599	0.576	0.622	0.134	0.569	0.467	0.637	34.9	25.4	35.6
SENSE-Skipgram	**0.678**	0.678	0.686	**0.374**	0.694	0.674	0.733	63.9	**33.8**	57.2

3.4.2.2 Datesets and Settings

We intrinsically evaluate our method on two standard tasks: the word similarity task by predicting the semantic similarity between words, and the word analogy task by predicting proportional analogies consisting of two pairs of words. The training corpus for all methods is a subset of the Wikipedia corpus, which contains 16 million words and 71,291 distinct words.

We conduct the word similarity task using the following benchmark datasets: **MC** (30 word-pairs) [97], **MEN** (3000 word-pairs) [98], **RG** (65 word-pairs) [99], **VERB** (143 word-pairs) [100], **WS** (353 word-pairs), and its similarity subset (**WS-sim**) and relatedness subset (**WS-rel**) [101]. Each word-pair in these benchmark datasets has a human-assigned similarity score. We calculate cosine similarity between the vectors of two words forming a test item, and report Spearman,Äôs rank correlation coefficient [102] between the rankings produced by the word embedding methods against the human rankings.

To assess the method,Äôs ability to perform semantic deduction, we evaluate word embedding methods using a word analogy task introduced by Mikolov et al. [8]. The task defines a comprehensive test that contains 19,544 questions divided into a semantic subset and a syntactic subset. The semantic subset contains five types of analogy ques-

tions about people or places, such as *"America is to New York as Australia is to ?"*. The syntactic subset contains nine types of analogy questions regarding verb tenses or forms of adjectives, such as *"good is to better as bad is to ?"*.

For each question, given w_1, w_2, w_3, it requires a fourth word w_4 to be generated to satisfy the question *"w_1 is to w_2 that is similar to w_3 is to w_4"*. The method we use to answer the question is by finding the optimal word using the following function:

$$v^* = \underset{v}{\operatorname{argmax}}\ cos(v, v_2) - cos(v, v_1) + cos(v, v_3), \tag{3.8}$$

where v_1, v_2, and v_3 are the embeddings of word w_1, w_2, w_3, and $cos(\cdot,\cdot)$ is the cosine similarity function. The best embedding of v^* is regarded as the answer.

3.4.2.3 Results

Table 31 shows the evaluation results for both the word similarity task and the word analogy task. From the results, we observe that:

(1) We observe that most KB-enhanced methods perform better compared to their baseline methods (e.g., Retrofit-CBOW v.s. CBOW), while the RCM method and the Jointreps method do not perform better than their corresponding baseline methods. This observation demonstrates that external KBs can boost the performance of word embeddings, but the methods of how to extract and model the semantic information may directly affect the performances. Our SENSE method significantly outperforms over all the baseline methods, which means that modeling semantic structures by concept convergence and word divergence is reasonable and effective.

(2) The SENSE method reports the best results in seven word similarity datasets and the word analogy dataset. In particular, the improvements reported by the SENSE method are statistically significant on MC, RG, WS, WS-rel, and WS-sim. We attribute the success of our method to its power in modeling structural information in the word embedding learning process.

(3) For the task of word analogy, the GloVe method is a much stronger baseline than the others. It is fair to say that the global counting information is more accurate for

Table 32: Evaluation results of multi-class text classification. Bold scores denote the SENSE method outperforms the corresponding baseline methods. Underlined scores are the best overall.

Methods	**Acc.**	**Prec.**	**Rec.**	**F1**
LDA	72.2	70.8	70.7	70.0
BOW	79.7	79.5	79.0	79.0
PV-DM	72.4	72.1	71.5	71.5
PV-DBOW	75.4	74.9	74.3	74.3
TWE	71.7	70.9	70.4	69.7
GloVe	62.3	61.2	61.1	60.5
CBOW	78.1	77.4	77.1	77.0
Skip-gram	80.2	79.6	79.1	79.0
Retrofit-CBOW	75.6	75.9	73.5	72.1
Retrofit-Skipgram	77.4	77.9	75.5	74.3
SENSE-CBOW	**81.4**	**80.8**	**80.3**	**80.2**
SENSE-Skipgram	81.7	81.2	80.6	80.6

semantic deduction compared to local co-occurrence information. The SENSE-Skipgram model still performs better than the GloVe method, demonstrating the generality and effectiveness of our method. It also implies that semantic structures are more reliable and stable knowledge than the relationship between word-pairs, and structural information can capture a word's latent relation in a global view.

3.4.3 Text Classification

We investigate the effectiveness of the SENSE method for text classification. The experiment is conducted on the 20NewsGroup[8] dataset. We use the bydate version which contains 18,846 documents from 20 different newsgroups. The dataset is separated into a training set of 11,314 documents and a test set of 7,532 documents. All documents are joined together as a corpus for training word embeddings. We tokenized the corpus with the Stanford Tokenizer[9] and convert it to lower case, then removed the stop words. The corpus is 30.4M and contains 6.3 million words.

We consider the following baselines, BOW, LDA, TWE [15], GloVe, Word2Vec, Retrofit

[8]http://qwone.com/ jason/20Newsgroups/.
[9]https://nlp.stanford.edu/software/tokenizer.shtml

and PV [103]. The BOW method represents each document as a bag of words and the weighting scheme is TFIDF (the top 50,000 words are selected). The LDA represents each document as its inferred topic distribution. We set the number of topics as 80. The PV method is an unsupervised learning algorithm that learns vector representations for documents by predicting words in the document, including distributed memory model (PV-DM) and the distributed bag-of-words model (PV-DBOW). For word embedding methods, we construct document embeddings $\mathbf{d}$ by simply averaging all word embeddings in the given document, i.e., $\mathbf{d} = \sum_{w \in d} \mathbf{x}_w$, where w is a word in document d, and $\mathbf{x}_w$ is the word embedding of word w. We regard document embedding vectors as a document feature and train a linear classifier using Liblinear[10], since the feature size ($d = 300$) is large, and the Liblinear can quickly train the linear classifier with high dimension features. The classifier is then used to predict the class labels of documents in the testing set. We report the macro-averaging accuracy, precision, recall, and F1−measure for comparison.

Table 32 shows the evaluation results of text classification. We observe that the SENSE-Skipgram method significantly outperforms all baseline methods, showing that our method better captures the semantic information of documents. Both SENSE-CBOW and SENSE-Skipgram outperform their basic methods, especially SENSE-CBOW achieves a 3.3% improvement over the CBOW method. Whereas two Retrofit methods do not perform as well as the basic Word2Vec method. This observation shows the superiority and generality of our SENSE method with modeling semantic structures in the word embedding learning process.

3.4.4 Query Expansion

We evaluate the performance of the SENSE method in query expansion for the information retrieval task. The experiment is conducted on the Reuters Corpus Volume 1 (RCV1) dataset, which contains 806,791 documents. We combine the *title* and *text* parts of all documents to construct a training corpus, and then tokenized the training corpus with

[10]https://www.csie.ntu.edu.tw/ cjlin/liblinear/

Table 33: Performance of different methods for query expansion on the RCV1 dataset. Bold scores denote that the SENSE method outperforms the corresponding baseline methods. Underlined scores are the best overall.

Methods	**P@10**	**P@20**	**MAP**	**F1**
BM25	44.6	44.1	40.8	41.5
TWE	55.4	49.5	44.2	43.5
GloVe	56.4	50.0	44.3	43.7
CBOW	56.4	49.1	44.3	43.8
Skip-gram	55.6	50.0	44.8	43.9
Jointreps	55.6	51.5	44.2	43.5
Retrofit-CBOW	57.6	50.8	44.3	43.6
Retrofit-Skipgram	56.6	50.4	44.8	43.8
SENSE-CBOW	58.4	51.9	45.1	44.2
SENSE-Skipgram	**58.2**	**50.6**	**45.0**	**44.1**

the help of the Stanford Tokenizer tool and convert every word to lower case. The corpus totals 16 million words.

The documents are divided into 50 collections, and each collection contains a training set and a test set. We implement the query expansion as follows: (1) we generated original queries by selecting the top 10 words in each collection, using the weighting scheme BM25; (2) then for each query q, we use word embeddings to select the top 5 most similar words with cosine similarity as its expansion words; (3) each expansion word w is associated with a weight as $w(q) * cos(\mathbf{q}, \mathbf{w})$, where $w(q)$ is the weight (BM25 score) of the original query, and $cos(\mathbf{q}, \mathbf{w})$ is the cosine similarity of the embeddings of the query and the expansion word. Finally, we construct an expanded query set Q^* which contains original queries and expanded words. Each query q in Q^* is associated with a weight, denoted as $w(q)$.

We retrieve the documents using the set Q^*. For each document d, its relevance score s to the query set is computed as $s = \sum_{q \in Q^*} f(q) * w(q)$, if $q \in d$, $f(q) = 1$; otherwise $f(q) = 0$. We report four standard evaluation metrics: the average precision of the top 10 documents (P@10) and top 20 documents (P@20), the mean average precision (MAP), and the F1−measure.

Table 33 reports the results achieved by the proposed method and the baselines. We

observe that all the query expansion methods significantly outperform the BM25 method, which indicates the effectiveness of employing word embeddings for query expansion. According to the table, the SENSE-CBOW method consistently outperforms all compared methods, and our methods significantly outperform their corresponding baseline methods. While other KB-enhanced method, i.e. Jointreps and Retrofit, perform slightly better than their baseline methods. Moreover, compared to the GloVe method and the TWE method, the SENSE method achieves remarkable improvements. This observation also indicates that semantic structures are more effective in capturing semantic features than collecting topical information and global co-occurrence information.

3.5 Summary

In this chapter, we proposed a novel approach for learning semantic structure-based word embedding, called SENSE. The proposed method can leverage transferable semantics from knowledge bases to improve word representations. The proposed method is a jointly word embedding learning method, incorporating the corpus and the knowledge base into capturing semantics of words. Our method differs from recent related work by constructing three-level semantic structures from the KBs, and by revealing concept convergence and word divergence to unit word's semantic granularity and abstraction. Experiment results with different datasets show that the proposed method outperforms the existing state-of-the-art word embedding learning methods on various tasks.

CHAPTER 4

DYNAMIC META-EMBEDDING WITH DOMAIN-SPECIFIC LATENT SEMANTIC STRUCTURES

4.1 Introduction

Word representation aims to capture the semantics of words based on their distributional properties in a large volume of natural language data and represents words in a computer-processed format such as fixed-length vectors [7, 8]. Recently, several distributed word representation methods have been developed to capture the meaning of words. The methods represent each word as a vector in a real-valued, low-dimensional space, in a way that embeds the semantics and syntactics of words using neural networks [8], or dimensional reduction on the word co-occurrence matrix [7].

Recent studies [31, 32] have observed significant variance in the quality and characteristics of the semantics captured by word representations generated by various existing methods according to the type of the training corpus, how the learning architecture is designed, and whether or not external knowledge is considered. These observations imply that different methods capture varied and complementary aspects of lexical semantics [21]. Moreover, a pioneering study from Yin and Schütze [22] found that a

combination of word representations from multiple methods leads to a better performance compared with an arbitrary individual method. They showed that it is feasible to combine the strengths of different word representations and thus yield a new representation space with improved overall expression quality. These novel observations address the importance of meta-embedding, which assembles different existing pre-trained word representations to yield a new and more powerful one.

Typically, meta-embeddings are derived from a set of source embeddings that are freely-available and pre-trained on large-scale corpora. To obtain meta-embeddings, most existing methods carry out straightforward mathematical operations over the set of source embeddings, such as concatenation [22], averaging [23], or constructing a new common embedding space by capturing complementary information in different source embeddings [24]. However, these methods treat different source embeddings with various qualities equally, and combine source embedding directly rather than adapting to the need of target tasks. To alleviate this problem, we propose to dynamically aggregate source embeddings with the attention schema to learn meta-embeddings.

Moreover, previous works did not consider the importance of knowledge from the target domain for meta-embeddings, which is crucial to downstream tasks. Therefore, there are two main problems in the learned meta-embeddings. Firstly, a semantic shift exists in the task domain and learning representation space. Naturally, the semantics of words vary across domains. For example, the semantics of *season* trained in a general domain might be closely related to *spring*, *summer*, *autumn*, and *winter*, but fail to be associated with a food flavoring which is specific to the kitchen domain. To alleviate the semantic shifts problem, we propose to explore the contextual information of words in the task domain, so as to guide the meta-embedding learning process. Secondly, the salient and specific words in the task domain are not well-characterized in the learned representation space. In natural languages, the distributions of salient and domain-specific words often determine the meaning of a document [33], but they are rarely appeared in other domains. As such, source domains often lack sufficient information for training high-quality embeddings for specific words. It is worth noting that using

contextual information alone does not capture the relationships of domain-specific words well, especially in low-resource scenarios. For example, consider the sentence *I own many KitchenAid appliances and love this brand*. Even though the context of *KitchenAid* can effectively capture its feature as a *brand*, it will lose some strong indicators with other specific words in the kitchen domain, such as *mixer* and *blender*. To alleviate this problem in our method, we exploit the relationship between salient and specific words in the task domain. More specifically, we construct a graph of salient words and explicitly organize their correlations, then employ graph convolution networks (GCNs) on the graph to transfer the latent structures into the learned meta-embedding space.

In brief, in this chapter, we propose an unsupervised and domain-specific meta-embedding approach, which goes a step further than previous ensemble methods. It dynamically leverages different source embeddings as background knowledge in an attention-driven strategy and mines specialized features of the task domain as domain-specific knowledge. To capture specialized knowledge in the task domain, we explore two ways to integrate meta-embeddings with domain-specific knowledge: 1) The contextual information discovered from the raw corpus, and 2) The domain-specific semantic structures conveyed by a graph built on salient words, where a stack of GCNs is applied over for in-depth knowledge mining.

4.2 Background

4.2.1 Meta-embedding Learning

With the design of various word representation methods, researchers have found that different methods vary significantly in the quality and characteristics of word semantics. Inspired by transfer learning methods [19], several works have shown that incorporating multiple word embeddings learned from different public corpora can improve performance in various NLP tasks. For example, Tsuboi [21] showed that using Word2Vec [8] and GloVe [7] embeddings together could improve the tagging accuracy. As such, meta-embeddings have become a popular trend, which combine existing high-quality pre-

trained word embeddings to produce more accurate and complete versions.

In an early attempt, Yin and Schütze [22] learned a projection layer that projects a word's meta-embedding to its source embeddings. Later, Coates and Bollegala [23] showed that the arithmetic mean of distinct source embeddings can yield meta-embeddings of a higher quality that are even better than the results from more complex meta-embedding learning methods. Bao and Bollegala [24] learned an auto-encoder as the meta-embedding space by accurately reconstructing all source embeddings simultaneously. Rather than reconstructing source embeddings, Bollegala et al. [104] proposed a locally linear meta-embedding method to reconstruct neighboring words in each source embedding space.

More recent ensemble methods have been designed to dynamically adapt source embeddings to task domains [105, 106]. For instance, Kiela et al. [107] explored the supervised learning of task-specific meta-embeddings, and applied their technique to sentence representations. Xu et al. [108] learned domain-specific meta-embedding as a lifelong learning framework, which extracts the context of words from past domains, to enrich the in-domain corpus and alleviate data scarcity problems. Hazem and Morin [109] explored a variety of embedding models and investigated their impact on the task of bilingual terminology extraction from specialized, comparable corpora. They showed that meta-embedding, based on specialized and general domain datasets, can improve performance when mining specialized bilingual lexicons. These methods typically align the source and target embeddings in a common space under the guidance of pivots, which are shared across domains.

4.2.2 Graph Convolution Networks in NLP

Graph convolution networks [110, 111] aim to extend standard convolution operations for general graph structures. GCNs have enjoyed wide success in the areas of computer vision [112, 113], demonstrating their general applicability and strong learning power for capturing structural information [114]. Only a few methods explore the applications of GCNs for NLP tasks such as machine translation [115] and event detection [116].

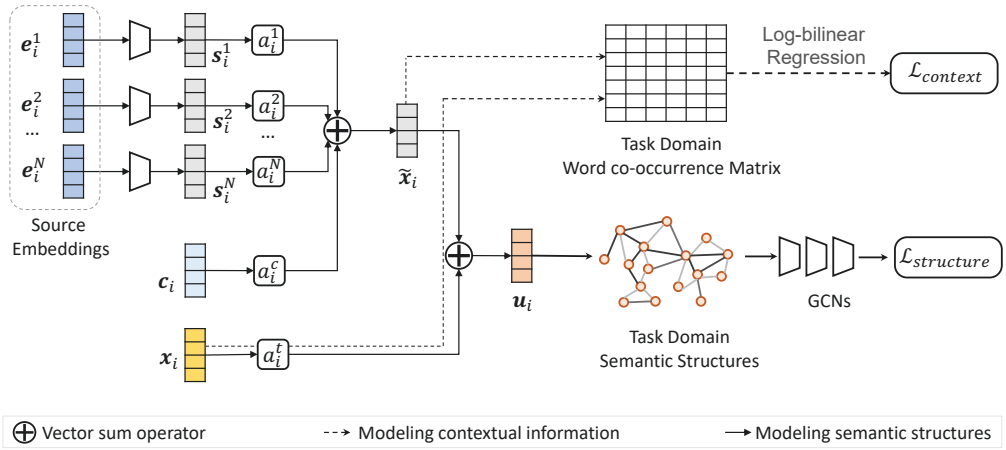


Figure 41: An overview of the proposed meta-embedding framework. The dashed lines denote the workflow of modeling contextual information, and the solid lines denote the workflow of modeling semantic structures.

Graph embeddings are also exploited for capturing semantics. For example, Liu et al. [117] proposed the use of graph-based semantic structures in knowledge bases under the assumptions of concept convergence and word divergence. Nguyen et al. [118] proposed learning for short-text semantic similarity with word embeddings and external graph-based knowledge sources.

In our method, GCNs are employed for learning domain-specific meta-embeddings. Here, semantic structures are captured by the graph of domain-specific words, where the nodes convey word semantics and the edges capture the correlations between words. GCNs are used to mine the latent structures of the graph, to capture domain-specific knowledge in the learned meta-embedding space.

4.3 Dynamical Meta-Embedding Method

Given a set of pre-trained source embeddings, our goal is to learn meta-embeddings for a task domain. A set of N source embeddings is denoted as $\{\mathbf{E}^n \in \mathbb{R}^{|\mathcal{V}_n| \times d_n}; 1 \leq n \leq N\}$, where $\mathcal{V}_n$ and d_n are the vocabulary and dimension, respectively. The output meta-embeddings are denoted as $\mathbf{U} \in \mathbb{R}^{|\mathcal{V}| \times d}$, where $\mathcal{V}$ is the vocabulary of the task domain and d is the embedding dimension. The overall framework of our method is illustrated in Fig. 41.

Table 41: Symbols and the descriptions. Matrix is denoted with bold capital letter, vector is denoted with bold lowercase letter, and scalar number is denoted with lowercase letter.

Symbol	Description	Symbol	Description
$\mathscr{C}$	Corpus.	$\mathscr{V}$	Vocabulary.
$\mathscr{G}$	Semantic graph.	$\mathscr{N}_i$	Neighbors of word w_i in $\mathscr{G}$.
$\mathbf{E}$	Source embedding matrix.	$\mathbf{e}_i$	Source embedding of word w_i.
d	Dimension of embedding.	$\mathbf{c}$,$\mathbf{x}$	The context embedding and the target embedding of word w_i.
$\mathbf{U}$	Meta-embedding matrix.	$\mathbf{u}_i$	Meta-embedding of word w_i.
$\mathbf{M}$	Co-occurrence matrix of $\mathscr{C}$.	$m_{i,j}$	Element of $\mathbf{M}$, co-occurrence times of word w_i and w_j.
$\mathbf{P}$	Adjacency matrix of $\mathscr{G}$.	$p_{i,j}$	Element of $\mathbf{P}$, edge weight of word w_i and w_j.
$\mathbf{A}$	Normalized symmetric adjacency matrix of $\mathscr{G}$.	$\mathbf{D}$	Degree matrix of $\mathscr{G}$.
$\mathbf{W}$, $\mathbf{w}$, $\mathbf{b}$	Trainable parameters.	a	Combination weight.

In this section, we first introduce how to integrate pre-trained source embeddings and generate meta-embeddings. Then, we detail how to guide the learning process to learn domain-specific meta-embeddings, considering both the contextual information and semantic structures. The used symbols and their descriptions are listed in Table 41.

4.3.1 Dynamically Combined Meta-embeddings

Given sufficient background knowledge of its context, humans can understand an unknown word. Inspired by this intuitive observation, we regard the pre-trained source embeddings as the background knowledge of words. To be specific, for each word w_i in the task domain, we first project its corresponding source embeddings $\{\mathbf{e}_i^n \in \mathbf{E}^n\}_{n=1}^N$ into a common d-dimensional space as follows:

$$\mathbf{s}_i^n = \mathbf{Z}^n \mathbf{e}_i^n + \mathbf{b}^n, \tag{4.1}$$

where $\mathbf{Z}^n \in \mathbb{R}^{d \times d_n}$ and $\mathbf{b}^n \in \mathbb{R}^d$ are the learnable linear projection parameters. If w_i is not in the source vocabulary $\mathscr{V}_n$, $\mathbf{s}_i^n$ is padded to the zero vector. We dynamically combine the projected source embeddings $\{\mathbf{s}_i^n \in \mathbb{R}^d\}_{n=1}^N$ as w_i's context embedding $\tilde{\mathbf{x}}_i$ with the self-attention mechanism as follows:

$$\tilde{\mathbf{x}}_i = a_i^c \mathbf{c}_i + \sum_{n \in [1,N]} a_i^n \mathbf{s}_i^n, \tag{4.2}$$

where a_i^* is the scalar combination weight computed by an attention layer. A randomly initiated vector $\mathbf{c}_i$ is introduced to capture the latent features when w_i works as a contextual word for other words in the task domain.

Then, we represent the specific semantics of w_i in the task domain using $\mathbf{x}_i \in \mathbb{R}^d$. The final meta-embedding $\mathbf{u}_i$ of w_i is computed as the dynamic combination of $\tilde{\mathbf{x}}_i$ and $\mathbf{x}_i$ as

$$\mathbf{u}_i = a_i^t \mathbf{x}_i + \tilde{\mathbf{x}}_i = a_i^t \mathbf{x}_i + a_i^c \mathbf{c}_i \ + \sum_{n \in [1,N]} a_i^n \mathbf{s}_i^n, \tag{4.3}$$

where a_i^t is a scalar weight computed through a self-attention mechanism. More specifically, together with weights in Eq. (4.2), we compute the combination weights with an attention layer applied with a softmax function:

$$[a_i^t, a_i^c, a_i^1, \cdots, a_i^N] = \text{softmax}(\mathbf{w}^\top [\mathbf{x}_i, \mathbf{c}_i, \mathbf{s}_i^1, \cdots, \mathbf{s}_i^N]), \tag{4.4}$$

where $\mathbf{w}$ is the learnable parameter, and the bias term is omitted for brevity.

It is notable that the underlying features of w_i in the task domain are conveyed by $\mathbf{c_i}$ and $\mathbf{x_i}$. Inspired by Pennington et al. [7], they are designed to represent the semantic characteristics of w_i as a context word and a target word, respectively. They are randomly initiated following a standard normal distribution and optimized in the learning process. Moreover, these two vectors can avoid the domain-specific words being represented by zero vectors when they are OOV (out-of-vocabulary) words in all source domains.

4.3.2 Domain-specific Knowledge

To capture the semantics of words in the task domain, we mine the domain-specific knowledge, then use this knowledge to guide the learning process of meta-embedding. Two kinds of knowledge with different granularities are considered, i.e., contextual information in the raw corpus and the underlying semantic structures.

4.3.2.1 Contextual Information

The contextual information of words in a corpus is the primary source of information available to all unsupervised methods for learning word representations. Following Pennington et al. [7], contextual information in the task domain is organized into a co-occurrence matrix $\mathbf{M} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ using a sliding window on the task domain corpus $\mathscr{C}$,

where the centre of the window is the target word and the other words are its context. The (i,j)-th element $m_{i,j}$ of $\mathbf{M}$ indicates the number of times word w_j appears in the context of word w_i. We use the log-bilinear regression model proposed by GloVe [7] to learn word representations, and the training objective is defined as:

$$\mathscr{L}_{\text{context}} = \frac{1}{2}\sum_i \sum_j f(m_{i,j})\left(\mathbf{x}_i^\top \tilde{\mathbf{x}}_j + b_i + \tilde{b}_j - log(m_{i,j})\right)^2, \tag{4.5}$$

where $m_{i,j} \in \mathbf{M}$, the central word is w_i and its contextual word is w_j, $\mathbf{x}_i$ is the defined specific representation of w_i, $\tilde{\mathbf{x}}_j$ is the context embedding of word w_j as defined in Eq. (4.2), and b_i and $\tilde{b}_j$ are real-valued scalar bias terms that compensate for the difference between the inner-product and the logarithm of the co-occurrence counts. The function $f(x)$ discounts the co-occurrences between frequent words and is given by:

$$f(m) = \begin{cases} (m/m_{\max})^\gamma & \text{if } m \le m_{\max}, \\ 1 & \text{otherwise}, \end{cases} \tag{4.6}$$

where $m_{\max} = 100$ and $\gamma = 0.75$, as suggested in Pennington et al. [7].

4.3.2.2 Semantic Structures

The context information is not sufficient to precisely and completely understand certain domain-specific words with specific meanings. This is mainly because the co-occurrence matrix $\mathbf{M}$ most likely considers all co-occurrences equally, which likely causes the domain-specific information deficiency problem, especially when the co-occurrence is rare. Thus, it fails to capture the rich relationships between words, especially for salient, domain-specific words. To alleviate this problem, we propose a GCN-based framework to further mine semantic structures among domain-specific words, which is essential for characterizing the task domain, and hence leads to more accurate meta-embeddings.

First, to identify domain-specific words, we adopt the term frequency-inverse document frequency (TF-IDF), which is an effective method for measuring the importance of words. The term frequency is the number of times a word appears in the document, and the inverse document frequency is the logarithmically scaled inverse fraction of the number of documents that contain the word. Then, we build an undirected graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$

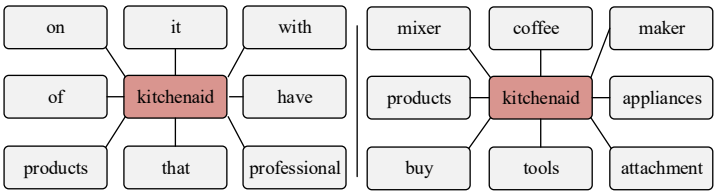


Figure 42: Top related words of *KitchenAid*, captured by the co-occurrence matrix **M** (left), and the domain-specific graph $\mathcal{G}$ (right). As seen, **M** and $\mathcal{G}$ address different levels of semantic context/structures. A comprehensive use of both leads to a more complete understanding of the meaning of words.

to capture domain-specific structures among salient words, where $\mathcal{V}$ is the node set of salient words and $\mathcal{E}$ is the set of edges to capture the relationships among salient words. Each edge connects two words, w_i and w_j, and the edge weight $p_{i,j}$ accounts for their relevance score $r_{i,j}$ and their individual importance scores, o_i, o_j, in the task domain:

$$p_{i,j} = \begin{cases} r_{i,j} \cdot (\frac{o_i + o_j}{2}) & \text{if } i \neq j, \\ \alpha \cdot o_i & \text{if } i = j, \end{cases} \tag{4.7}$$

where $\alpha = 1$ is a hyper-parameter, which will be quantitatively evaluated in Section 4.5. The relevance score $r_{i,j}$ of two words w_i and w_j is defined as their co-occurrence times in a sentence or document. The importance scores o_i, o_j are obtained through TF-IDF.

For an intuitive comparison, considering a salient word *KitchenAid* in the kitchen domain, Fig. 42 shows the most related words obtained from the co-occurrence matrix **M** (left), and the domain-specific graph $\mathcal{G}$ (right). As can be observed, **M** focuses more on local context (due to the use of a local sliding window), while $\mathcal{G}$ is able to capture global semantic relations among significant words.

To encode the structures of graph $\mathcal{G}$ in the learned meta-embedding space, we apply the effective and powerful GCN over $\mathcal{G}$ to mine valuable domain-specific knowledge. When adapting one single GCN layer [111], we have:

$$\mathbf{H} = \sigma(\mathbf{AUW}), \tag{4.8}$$

where $\mathbf{U} \in \mathbb{R}^{|\mathcal{V}| \times d}$ is the input embedding matrix, its element $\mathbf{u}_i$ is the input embedding of w_i which is computed using Eq (4.3)[1], $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}| \times d}$ is the output hidden representation of

[1]Noting that the **U** here is the meta-embedding matrix with random initial parameters.

the nodes, $\mathbf{A} \in [0,1]^{|\mathcal{V}|\times|\mathcal{V}|}$ is the normalized symmetric adjacency matrix of $\mathcal{G}$, $\mathbf{W} \in \mathbb{R}^{d\times d}$ is a learnable weight matrix, and $\sigma(\cdot)$ indicates a non-linear activation function (e.g., *ReLU*). Here, $\mathbf{A}$ is computed as:

$$\mathbf{A} = \mathbf{D}^{-\frac{1}{2}}\mathbf{P}\mathbf{D}^{-\frac{1}{2}}, \tag{4.9}$$

where $\mathbf{P}$ is the adjacency matrix of $\mathcal{G}$ with a self-loop. $\mathbf{D}$ is the degree matrix, i.e., $d_{i,i} = \sum_j p_{i,j}$, $p_{i,j} \in \mathbf{P}$.

Eq. (4.8) can be represented in an alternative form:

$$\mathbf{h}_i = \sigma(\textstyle\sum_{w_j \in \mathcal{N}_i} \mathbf{W}\mathbf{u}_j), \tag{4.10}$$

where $\mathcal{N}_i = \{w_j : p_{i,j} > 0\}$ indicates neighbors of w_i. The bias term is omitted for brevity. Due to the self-loops (Eq. (4.7)), we have $w_i \in \mathcal{N}_i$ and the input feature $\mathbf{u}_i$ of w_i affects its induced representation $\mathbf{h}_i \in \mathbb{R}^d$. In addition, as GCN operates on the neighbors of a node, the updated node representation $\mathbf{h}_i$ efficiently captures semantic structures over $\mathcal{G}$.

One single GCN layer only encodes information about immediate neighbors. By stacking multiple GCN layers, higher-degree neighborhoods can be incorporated thus capturing more complex semantic structures/correlations:

$$\mathbf{h}_i^{k+1} = \sigma(\textstyle\sum_{w_j \in \mathcal{N}_i} \mathbf{W}^{k+1}\mathbf{h}_j^k), \quad \mathbf{h}_i^0 = \mathbf{u}_i. \tag{4.11}$$

The training objective is designed to predict a target word w_i given its neighbors $\mathcal{N}_i$ over $\mathcal{G}$, i.e., to maximize the following energy:

$$\textstyle\sum_{w_i \in \mathcal{V}} log Pr(w_i | \mathcal{N}_i). \tag{4.12}$$

The training loss is thus defined as:

$$\mathcal{L}_{\text{structure}} = -\textstyle\sum_{w_i \in \mathcal{V}} \left(\mathbf{u}_i^\top \mathbf{h}_i^k - log \sum_{w_j \in \mathcal{V}} exp(\mathbf{u}_j^\top \mathbf{h}_i^k)\right), \tag{4.13}$$

where $\mathbf{u}_i \in \mathbb{R}^d$ is the final meta-embedding of w_i. To encourage the final meta-embedding to capture multi-order semantic dependency structures among domain-specific words, we improve Eq. (4.13) to:

$$\mathscr{L}_{\text{structure}} = -\sum_{k=0}^{K} \sum_{w_i \in \mathcal{V}} \left(\mathbf{u}_i^{\top} \mathbf{h}_i^k - log \sum_{w_j \in \mathcal{V}} exp(\mathbf{u}_j^{\top} \mathbf{h}_i^k) \right). \tag{4.14}$$

When $K = 0$, the final meta-embedding $\mathbf{u}_i$ is only required to be consistent with the initial input $\mathbf{v}_i$. When $K = 1$, the GCN encodes additional information from immediate neighbors into $\mathbf{u}_i$. When K is further increased (i.e., more GCN layers are stacked), $\mathbf{u}_i$ will encode K-order neighborhoods (i.e., information about nodes at most K hops away), but with more risk of contamination from irrelevant words. In our implementation, we set $K = 2$. The validation for this can be found in Section 4.5.

4.3.3 Learning Process

We formulate the joint objective as a minimization problem as follows:

$$\mathscr{L} = \mathscr{L}_{\text{context}} + \lambda \mathscr{L}_{\text{structure}}, \tag{4.15}$$

where $\lambda \in \mathbb{R}^+$ is a non-negative real-valued combination hyper-parameter. To optimize the aforementioned model, we follow the GloVe method to minimize $\mathscr{L}_{context}$. We follow the GCN method [111] to optimize the $\mathscr{L}_{structure}$.

The overall algorithm for learning our meta-embeddings is listed in Algorithm 2. During training, we use stochastic gradient descent (SGD), with learning rates scheduled by AdaGrad [119]. For the efficiency of the algorithm, it is mainly affected by the number (K) of GCN layers. In our experiments, we set K=2 and evaluate the effectiveness of our method with different K in Section 4.5.1.

4.4 Experiments

In this section, we conduct comprehensive experiments to verify the effectiveness of our method. We first detail the implementation of the proposed domain-specific meta-

Algorithm 2 Meta-Embedding Algorithm

Input: A set of source word embeddings $\{\mathbf{E}^n \in \mathbb{R}^{|\mathcal{V}_n| \times d_n}\}_{n=1}^{N}$, and the task domain corpus $\mathscr{C}$.
Parameter: Dimensional d of word embeddings, the maximum number of iterations T.
Output: Each word's meta-embedding $\mathbf{u}_i \in \mathbb{R}^d$ in task domain.

1: Initialize $\mathbf{c}$ and $\mathbf{x}$ for each word, and generate the word's context embedding $\tilde{\mathbf{x}}$.
2: Construct the co-occurrence matrix $\mathbf{M}$ and domain-specific graph $\mathscr{G}$
3: Project source embeddings into a common d-dimension space for each word w_i using Eq. (4.1)
4: **while** 1 to T **do**
5: Optimize a, $\mathbf{c}$, and $\mathbf{x}$ according to the semantic structures with $\mathscr{L}$ in Eq. (4.15)
6: **end while**
7: **return** $\mathbf{u}_i$ of each word w_i

embeddings, including the used source embeddings and the construction of the co-occurrence matrix and domain-specific graph. Then, we describe the used baseline methods. Following this procedure, we report the experiments on two tasks, i.e., text classification and relation extraction.

4.4.1 Implementation

Source Embeddings: Three widely-used pre-trained word embeddings are used as the source embeddings of our method:

- **CBOW**: Trained by continuous bag-of-words method and released by Mikolov et al. [8], CBOW has a vocabulary of 929,019 words. The dimension size is 300, the training corpus is Google News with 100 billion tokens.

- **GloVe**: The most widely used pre-trained word embeddings. GloVe is trained by Pennington et al. [7] using global word co-occurrences information in the corpus. There are 1,193,514 words in the vocabulary, and the dimension size is 300. The training corpus is web-crawled text with 42 billion tokens.

- **fastText**: Released by Mikolov et al. [120]. The pre-trained word embeddings of fastText are equipped with flexible sub-word structures. It is trained on the

Table 42: Statistics of the corpora used in our experiments. #Vocab is the size of vocabulary. Avg.Len. is the average length of documents in the corpus.

Dataset	**#Train**	**#Test**	**#Vocab.**	**#Tokens**	**Avg.Len.**	**Size**
Kitchen	1,600	400	9,066	140K	69.33	711KB
DVD	1,600	400	16,018	172K	85.33	903KB
Electronics	1,600	400	9,641	136K	67.27	697KB
Book	1,600	400	16,575	176K	87.07	951KB
CR	3,450	320	5,319	74K	17.70	353KB
TREC	5,452	500	8,604	54K	9.71	264KB

Wikipedia corpus with 16 billion tokens, with a vocabulary size of 1 million and dimensionality of 300.

In our experiments, the dimensionality of the learned meta-embeddings (d) is set to 300, and the evaluation of different dimensionally is detailed in Section 4.5. For the co-occurrence matrix $\mathbf{M}$, we set the context window to be the 10 tokens preceding and succeeding a given word in a sentence. We extract unigrams from the co-occurrence windows as the corresponding context words. We down-weight distant (and potentially noisy) co-occurrences using the reciprocal $\frac{1}{l}$, where l is the distance in tokens between two words that co-occur. When constructing the specific graph $\mathcal{G}$, we set each sentence as a window. In each window, stop-words are discarded and only the correlations among the top-10 most important words are counted (to reduce noise).

4.4.2 Baselines

For a comprehensive comparison, the proposed method is compared against state-of-the-art baselines from three classes:

1) Pre-trained source embeddings. Three state-of-the-art pre-trained word embeddings trained from large-scale corpora, namely CBOW, GloVe, and fastText, are compared in the experiments as detailed in Section 4.4.1.

2) Domain-specific embeddings. We train the domain-specific word embedding on the task domain corpus, using the Word2Vec and GloVe methods, denoted as CBOW^t,

Skipgram^t, and GloVe^t, respectively. We use the the official public tools with the default settings. The dimensionality is also set to 300.

3) Meta-embedding methods. We compare the proposed method with the four following state-of-the-art meta-embedding methods:

- **1ToN** [22], which learns meta-embeddings for the words in an intersection vocabulary, by predicting their representations in the individual source embedding sets. 1ToN's extension, **1ToN+**, which is designed for handling OOVs, is also included in our experiments.

- **LLE** [104], which generates the meta-embedding space by learning a locally-linear projection that predicts the words' neighborhoods in the source embeddings.

- **AEME** [24] trains an auto-encoder to encode source embeddings into a common meta-embedding space and then decodes them back to the original source space.

- **REGrep** [105] is a domain-specific word embedding method. It is trained on an in-domain corpus with a regularizer that constrains common words to be similar to their source embeddings.

We use the released embeddings of 1ToN and 1ToN+. LLE, AEME, and REGrep, which are trained using their codes published online with the same source embeddings as detailed in Section 4.4.1. Moreover, we train an extended version of these meta-embedding methods, denoted as LLE+t and AEME+t, by integrating CBOW^t trained on the task domain as an extra source embedding.

4.4.3 Task I: Text Classification

The first evaluation task is conducted on text classification over three datasets:

- **Amazon Reviews** [121]. The dataset consists of product reviews in Amazon with labels to classify the review as positive or negative. Its four subsets are conducted in our experiments, i.e., **Kitchen**, **DVD**, **Electronics**, and **Book**.

Table 43: Overall performance for the text classification task, conducted on four subsets (i.e., *Kitchen, DVD, Book*, and *Electronics*) of *Amazon Reviews*, *Custom Reviews* (CR), and *TREC* datasets. The highest scores are marked in bold.

Methods	SVM classifier						CNN classifier					
	Amazon Reviews				CR	TREC	Amazon Reviews				CR	TREC
	Kitchen	DVD	Book	Elec.			Kitchen	DVD	Book	Elec.		
CBOW	82.00	79.00	81.50	83.75	81.56	78.20	81.53	69.25	69.63	77.43	82.72	85.28
GloVe	82.75	79.75	81.25	81.00	82.81	82.20	85.90	78.68	78.68	81.25	84.63	87.02
fastText	83.25	**80.75**	80.00	81.75	81.87	81.80	79.43	67.38	68.58	75.28	80.50	84.90
CBOW^t	81.25	73.50	74.50	76.50	80.94	89.40	85.08	80.28	79.38	82.68	78.44	89.58
Skipgram^t	82.00	73.75	73.75	76.75	80.25	89.64	85.88	80.65	79.03	82.90	78.63	89.88
GloVe^t	78.50	68.00	68.75	71.25	73.13	70.20	83.85	75.10	73.43	79.58	75.97	83.78
1ToN	76.25	72.25	77.25	77.50	81.50	81.70	84.80	78.25	79.15	83.73	76.56	82.98
1ToN+	73.00	71.25	78.25	73.50	68.50	77.20	84.85	79.75	80.30	84.25	76.81	80.84
LLE	76.50	71.25	75.00	73.25	82.50	81.70	71.95	77.03	75.23	72.93	73.13	72.60
LLE+t	77.75	73.25	78.50	74.75	83.50	68.00	74.60	76.05	74.40	74.43	76.13	79.30
AEME	79.50	75.50	79.50	76.25	84.20	77.80	79.09	76.09	78.41	71.38	78.53	81.83
AEME+t	82.00	80.00	78.75	78.25	82.50	76.80	78.68	75.53	77.54	73.98	78.78	82.02
REGrep	81.00	77.80	77.80	78.00	75.60	87.80	82.15	79.23	78.80	79.38	83.28	90.20
Our method	**84.75**	**80.75**	**84.00**	**84.50**	**84.69**	**92.30**	**88.15**	**82.30**	**82.50**	**85.85**	**86.59**	**92.78**

- **Custom Reviews** (CR) [122]. The dataset contains various products (such as cameras) with labels to classify the review as positive or negative.

- **TREC** [123]. This is a question-type classification dataset, which coarsely classifies the question sentences into six types.

All the datasets are organized following their standard splits. The statistics of the used datasets are listed in Table 42. To conduct the experiment, two types of classifiers are used in our experiments:

1) **SVM**: Each document is represented as a bag-of-words, and we compute the centroid of the embeddings for each bag to represent the document. Then, we train a Linear SVM classifier using the training portion of each dataset and evaluate the classification accuracy using the corresponding testing portion.

2) **CNN**: A CNN (Convolutional Neural Network) based classifier is trained with 100 filters, each of size 5, for max-pooling. The initial learning rate is set to 0.01, and the batch size is set to 50. The accuracy score averaged over ten repetitions is reported.

Table 43 lists the overall performances of the text classification task. The results show that our method significantly outperforms all other competitors across different settings. The improvement of our proposed meta-embedding method is statistically significant

Table 44: The comparison of our method with the pre-trained language models.

Methods	**Amazon Reviews**				**CR**	**TREC**
	Kitchen	DVD	Book	Elec.		
Our Method-SVM	84.75	80.75	84.00	84.50	84.69	92.30
Our Method-CNN	88.15	82.30	82.50	85.85	86.59	92.78
BERT-base	92.50	90.25	87.75	90.25	93.75	96.20
RoBERTa-base	93.50	91.00	92.25	92.50	95.63	96.80

over the best single source embeddings, indicating the effectiveness of the technique. More specifically, the domain-specific embeddings, i.e., CBOW^t and GloVe^t, achieve poor performances compared with the source embeddings, especially on the Amazon Reviews dataset with only 2,000 reviews. The main reason for the poor performance is that the task domain corpus is too small to train reliable word representations, which suggests the importance of using extra knowledge from source embeddings to enrich the semantics of in-domain words.

Table 43 demonstrates that our method outperforms the other meta-embedding methods, indicating the advantage of our domain-specific meta-embeddings. Furthermore, our method achieves significant improvements over the meta-embeddings, which do not consider information in the task domain, such as 1ToN, 1ToN+, LLE and AEME. Both LLE+t and AEME+t outperform their basic methods, LLE and AEME, by 1.9% and 0.56%, respectively, indicating the significance of task domain information. These observations demonstrate the importance of adapting the learned embedding to the specific domain. Notably, one reason for the poor performance of LLE and AEME is that they suffer from 16.9% OOVs on average, for all the datasets. Our method alleviates this problem by introducing a learnable vector $\mathbf{c}$, thus achieves better performances. Our method achieves substantial gains compared with meta-embeddings which contain domain-specific information, i.e., LLE+t, AEME+t, and REGrep. For example, compared with the second-best method REGrep, our method surpasses it by 4.8% on average. This indicates that our proposed approach is more effective in the exploration of domain-specific knowledge.

In Table 44, we compare our method with classifiers build on pre-trained language

Table 45: Overall performance on the relation extraction task.

Methods	**P@100**	**P@200**	**P@300**
CBOW	78.00	73.50	66.00
GloVe	80.00	68.00	64.33
fastText	81.00	72.00	67.00
CBOW^t	77.00	74.50	67.33
Skipgram^t	78.00	73.50	67.37
GloVe^t	78.00	70.00	63.33
1ToN	78.00	76.00	68.33
1ToN+	75.00	71.00	66.67
LLE	76.00	70.50	66.00
LLE+t	81.00	72.00	69.33
AEME	78.00	73.50	66.00
AEME+t	82.00	72.50	65.33
Our method	**86.00**	**77.50**	**72.30**

models, i.e., BERT-base [10] (the uncased version) and RoBERTa-base [11]. In our experiment, the sequence length is set to 128, the learning rate is searched in range of {1e-5, 2e-5, 3e-5}, the training epoch is set to 3, and the batch size is set to 32. We observe that the pre-trained language models achieve better results. The main reasons are two-fold: (1) the pre-trained language models are contextual representation method which can cover the long-term contextual information; (2) the pre-trained language models are trained on large-scale textual data using huge-amount computing resources, which convey rich linguistic knowledge and background knowledge. While our method is a distributed representation method, which is a lightweight and easy to be implemented approach. As such, it is unfair to compare our method with pre-trained models directly. Compared with other distributed methods, our method achieves better results and highlights the importance of domain-specific knowledge, which is also informative to improve the contextual representation methods.

4.4.4 Task II: Relation Extraction

We evaluate the proposed method using the relation extraction task on the New York Times corpus (NYT) dataset[2], developed by Riedel et al. [124]. This dataset was gener-

[2]http://iesl.cs.umass.edu/riedel/ecml/

ated by aligning Freebase relations with the New York Times corpus. Entity mentions are found using the Stanford named entity tagger [125] and are further matched to the names of Freebase entities. The Freebase relations are divided into two parts, one for training and one for testing. To be specific, sentences from the years 2005-2006 are combined to form the training set, while sentences from 2007 are used for testing. There are 53 possible relationships, including a special relation *NA*, which indicates there is no relation between head and tail entities. The training data contains 522,611 sentences, 281,270 entity pairs and 18,252 relational facts. The testing set contains 172,448 sentences, 96,678 entity pairs and 1,950 relational facts. We use Piecewise Convolution Neural Networks (PCNNs) [126] as the distant supervised relation extraction model. In the experiments, different word representations are used to initialize the model. Precision@100, 200, and 300 (P@N) are reported.

The overall evaluation results are reported in Table 45. It is shown that:

- For the relation extraction task, our method also achieves a promising performance, with an average improvement of 6.4% over source embeddings and 5.8% over the meta-embeddings methods. The pre-trained source embeddings achieve better performance than domain-specific embeddings which are only trained on the task domain corpus, i.e., pre-trained GloVe beats GloVE^t by 2% in terms of P@100. This observation shows that the pre-trained source embeddings are useful and efficient resources for capturing word's semantics.

- The methods LLE+t and AEME+t achieve better results than their source embeddings, as well as their corresponding methods, LLE and AEME. These results suggest that task-domain information might be useful for guiding the integration of source embeddings. Our method jointly considers both the background knowledge from source embeddings and the domain-specific knowledge from the task domain.

- Our method consistently outperforms LLE+t and AEME+t, which also considers the domain-specific information. This performance is attribute to our GCN-based framework, which can capture more intricate correlations among domain-specific

Table 46: Evaluation of essential components and different GCNs.

Method	Amazon Reviews			
	Kitchen	DVD	Book	Elec.
Full model ($K = 2$)	**84.75**	**80.75**	**84.00**	**84.50**
w/o task domain semantic structures	82.25	78.75	81.75	81.25
w/o task domain contextual information ($K = 2$)	74.25	73.75	72.50	74.50
w/o source embeddings ($K = 2$)	76.50	77.75	77.50	78.00
Full model ($K = 0$)	83.25	79.75	82.00	83.25
Full model ($K = 1$)	83.50	80.50	83.50	83.75
Full model ($K = 3$)	83.00	80.25	83.75	83.25
Full model ($K = 4$)	82.75	78.50	81.50	83.00

words, showing the superiority and generality of our technique for modeling domain-specific information in the word embedding learning process.

4.5 In-depth Analyses

In this section, we qualitatively and quantitatively investigate the effects of essential components in the proposed method, assess the impact of key parameters, and visualize the attentions. The experiments are conducted on the Amazon Reviews dataset.

4.5.1 Ablation Study

We assess the contribution of each component of the proposed method. One by one, each component is removed, and the test accuracy on the Amazon Review dataset. The SVM classifier is used for comparison. As shown in Table 46, our full meta-embedding method performs better than other variants when only task domain context or semantic structures are considered. This observation indicates that the domain-specific information is useful to improve the quality of meta-embeddings, either the contextual information from the raw corpus or the latent semantic structures conveyed by the graph of salient words. In our method, to better capture a word's context features, we explore background knowledge ($\{\mathbf{s}_i^n\}_{n=1}^N$ from source embeddings (Eq. (4.2)). To validate the effectiveness of source embeddings, we provide a baseline, *w/o source embeddings*, which does not use

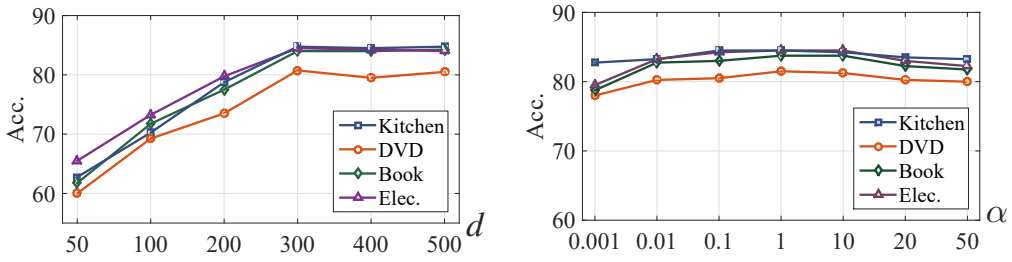


Figure 43: Performance for parameter selection on text classification task for meta-embedding dimension d and α.

pre-trained embeddings. A significant drop in performance is observed, indicating that the source embeddings indeed provide useful background knowledge for capturing word's semantics in the embedding space.

Moreover, we verify the effectiveness of GCNs for capturing the latent semantic structures of the task domain. In our method, we capture multi-order relations among domain-specific words by stacking K GCN layers. To evaluate the influence of the GCN design, we report the performance as functions of a variety of Ks. As shown in Table 46, the performance increases when more semantic structures are explored ($K\uparrow$). However, when more than three GCN layers are stacked, the performance becomes worse, due to disturbance from irrelevant words.

4.5.2 Parameter Validation

We investigate the influence of the dimensional d of the meta-embeddings and the impact of parameter α in Eq. (4.7). The performance with different values of d (i.e., $d \in \{50, 100, 200, 300, 500\}$) is shown in Fig. 43. We see a gradual improvement in performance with increasing d, which plateaus after $d = 300$, which indicates that adding new dimensions over 300 does not result in more performance improvement. Next, we report the performance for $\alpha = \{0.001, 0.01, 0.1, 1, 10, 20, 50\}$. As shown in the right of Fig. 43, $\alpha = 1$ achieves the best performance. The recommended settings for d and α is 300 and 1, respectively.

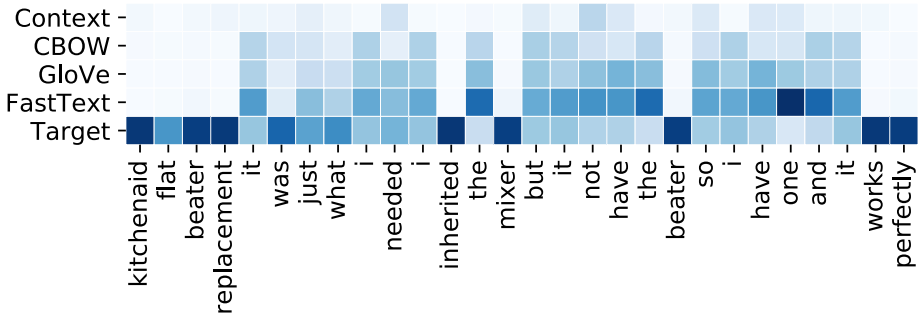


Figure 44: Visualization of self-attention-based combination weights (a deeper color signifies a higher weight). *CBOW*, *GloVe*, and *fastText* are the pre-trained source embeddings. *Context* denotes **c**. *Target* denotes **x**.

Table 47: Top five words related to *season* in different embedding spaces.

fastText	**CBOW**^{t}	**LLE**	**AEME**	**Our method** (w/o semantic structures)	**Our method** (w semantic structures)
year	remembering	week	finale	all-clad	spices
finale	specifications	weekend	year	starbucks	sauce
off-season	record	finals	league	unpredictable	flavor
winter	holiday	playoffs	postseason	stones	ingredients
summer	load	doubleheader	playoffs	gourmet	marinade

4.5.3 Weight Visualization

To gain an intuitive understanding of what happens when source embeddings are dynamically combined (Eq. (4.3)), we visualize the attention-based combination weights (a defined in Eq. (4.4)). As shown in Fig. 44, general words, such as *needed* and *have*, receive evenly divided attention across different embeddings. The method fastText is given relatively higher weights, possibly because it is trained on a more recent version of Wikipedia, and it is more favored for text classification. In contrast, domain-specific words, such as *KitchenAid*, *beater* and *mixer*, are more reliant on the target embedding **x** (denoted as *Target* in the last row), which conveys more structural information in the task domain.

4.5.4 Case Study

For an in-depth analysis, we conduct a case study to evaluate the proposed method qualitatively. To be specific, we consider the word *season* in the kitchen domain and show its predicted nearest neighbors in different embedding spaces. As shown in Table 47, the pre-trained source embedding (i.e., fastText) and meta-embeddings (i.e., LLE and AEME) predict the most closely related words as being *year*, *summer*, *holiday*, *week*. They are prone to capture the sense of *season* as one of the four periods of the year, which is the main semantics used in the general domain. However, the task domain is the kitchen domain. The meaning of *season* is to heighten or improve the flavor of food by adding condiments, spices, herbs, or the like. Being well-deserving of the domain-specific graph which provides more hints, our method reveals its domain-specific semantics. As shown in the table, our method is able to capture its strong relationship with *spices*, *sauce*, and *flavor*.

4.6 Summary

In this chapter, we proposed an unsupervised meta-embedding method, that dynamically integrates pre-trained source embeddings to adapt the task domain by exploring background knowledge from source domains and domain-specific knowledge mining from both the word distribution in the corpus and correlations among significant words. The domain-specific words and their co-occurring information is used to construct the graph of the domain structure, and the GCN-based model is designed to explicitly preserve such structure consistency in the learned embedding space. Experiments on different tasks and in-depth analyses show that our method is able to produce accurate meta-embeddings of words, which efficiently address domain-specific knowledge.

CHAPTER 5

PIVOT-BASED CROSS-LINGUAL RETRIEVAL WITH LIMITED BILINGUAL RESOURCE

5.1 Introduction

Word representation methods map words in a latent vector space where words with similar meanings are close to each other. Researchers [35] have shown that word representations trained for different languages can be aligned in a same vector space using a small-size dictionary. In the aligned embedding space, words with same meaning from different languages are close to each other. The aligned multilingual word representations are widely used for cross-lingual tasks, such as cross-lingual entity linking [25] and cross-lingual text classification [127]. However, multilingual word representations are good at encoding word-level semantics from different languages, but they have limited representation ability to encoding sentences or documents. To alleviate this problem, multilingual pre-trained language models (such as multilingual BERT) have powerful contextual representation ability. But, they are not available for large-scale retrieval due to efficiency problems.

In this chapter, we aim to bridge the cross-lingual semantic gap for large-scale cross-lingual retrieval. We focus on the entity linking task [128], which associates mentions in a sentence with their corresponding entities in a knowledge base. Considering the diversity of languages used on the web, cross-lingual entity linking (XEL) [26, 129] where the sentences are in a source language different from the knowledge base language has attracted wide attention recently. XEL is an important component task for many downstream tasks, such as cross-lingual knowledge-based question answering [130], cross-lingual information extraction [131], etc.

Typically, XEL consists of two steps: (1) *candidate retrieval*, which retrieves a small subset (e.g. 1000) of plausible candidates from a large set of KB entries in the target language (e.g. 6 million English entities in DBPedia); and (2) *entity disambiguation*, which re-ranks the selected candidates and returns the most likely entities. Candidate retrieval plays a critical role for cross-lingual entity linking, since missing entities in this step will never be recovered by the downstream disambiguation step. Nevertheless, the quality of candidate retrieval under a cross-lingual setting is far from complete. For example, as illustrated in Zhou et al. [129], a recall of retrieved candidates can reach over 80% for English mentions with the help of a Wikipedia mention-entity dictionary, while that of the state-of-the-art method is only 40% for mentions in Telugu (a Dravidian language spoken in southeastern India). The low-quality of the candidate retrieval step is gradually becoming a key obstacle in the XEL task.

In general, candidate retrieval for monolingual entity linking suffers from *mention-entity gap*, because surface forms of entities often differ from mentions. For example, the mention *Einstein* is linked to the entity `Albert_Einstein`. For XEL tasks, candidate retrieval is also hindered by *cross-lingual gap*, since the source and target languages are in different scripts. For example, *Manhattan Bridge* refers to *pont de Manhattan* in French. To fill these two gaps, existing works mainly take two types of approaches: lexicon-based and semantic-based approaches.

The lexicon-based approach usually creates lexicons to bridge both gaps with Wikipedia resources. For example, Pan et al. [34] proposed to map source-language mentions to

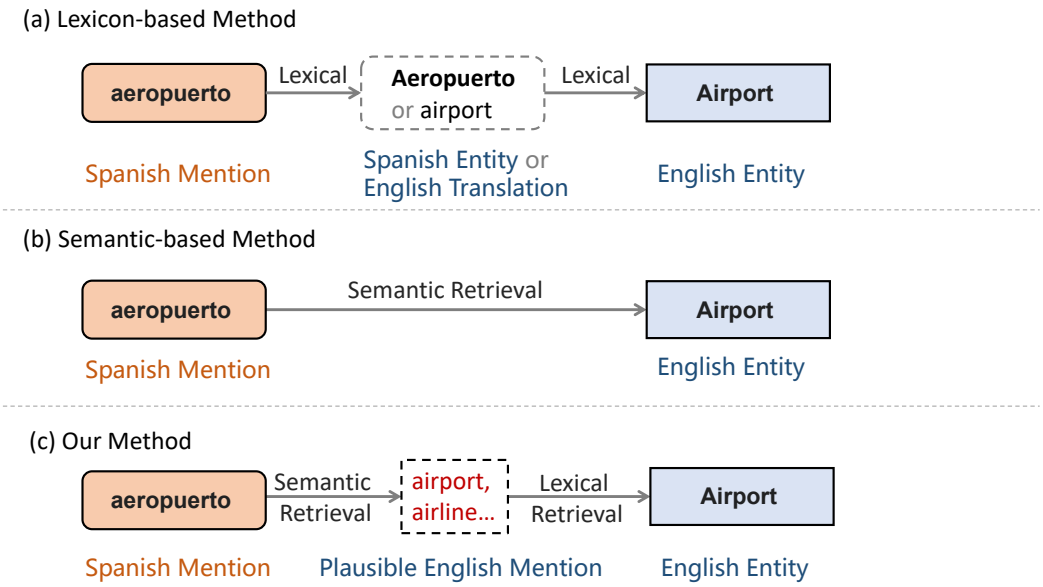


Figure 51: Comparison of lexicon-based method, semantic-based method, and our pivot-based method.

target-language ones using inter-language links, and then retrieved candidate entities from an English mention-entity dictionary. A major problem with such approach is that it heavily relies on Wikipedia inter-language links, which can only cover a small percentage of target-language entities, and this problem is especially severe for low-resource languages.

The semantic-based method generates candidate entities by leveraging the cross-lingual semantic retrieval. It usually builds an aligned embedding space between the source-language mentions and target-language entities, where synonyms of different languages have similar embeddings, and candidate entity retrieval can be undertaken by searching nearest neighbors of each mention in the embedding space of target entities. However, a single low-dimensional embedding has limited representation capacity for mentions or entities, and tends to lose lexical matching information which is critical to retrieval [132]. For example, for the French mention *pont de Manhattan* (*Manhattan Bridge* in English), the semantic-based approach tends to retrieve different kinds of bridges, such as *Belmont Avenue Bridge in Philadelphia, Bridges of the Merritt Parkway*, which successfully captures the keyword *bridge* while ignores the other one *Manhattan*.

In this work, we propose a pivot-based approach for the cross-lingual entity candidate retrieval task, which fully explores the advantages of both lexicon-based and semantic-based retrieval and avoids their limitations. In one aspect, it is usually difficult to derive an inter-language lexicon with high coverage. However, there is relatively large volume of monolingual data for both source and target languages, which can be fully leveraged by pre-trained models to map words into embeddings. Furthermore, with only a small set of bilingual word pairs, cross-lingual alignment can easily map word embeddings from one language to those in another language. Therefore, our approach first converts source-language mentions into an intermediary set of plausible target-language mentions with word-level cross-lingual semantic retrieval and a selective mechanism. In another aspect, there is usually rich lexicons such as alias or anchor texts to bridge the gap between entities and mentions in the target language. Therefore, our approach further conducts lexical retrieval with the generated intermediary target-language mentions.

We illustrate the difference among lexicon-based, semantic-based, and our pivot-based approach in Figure 51. Compared to lexicon-based approach, the proposed pivot-based method does not rely on Wikipedia inter-language links, and it fully leverages pre-trained word embeddings and only needs a small set of seed bilingual word pairs to learn cross-lingual alignment. Compared to semantic-based approach, our method converts a source-language mention into a diverse intermediary set of plausible target-language mentions with a flexible selective mechanism, and fully leverages the rich lexical resources of target-language knowledge base, and thus can retrieve more diverse and accurate candidates.

We evaluate the proposed method on two XEL entity linking datasets, QALD which contains non-Wikipedia questions in 8 languages and WIKI-LRL which contains Wikipedia titles in 3 low-resource languages. Experimental results show that it outperforms both the lexicon-based and semantic-based approach by a large margin. The source code of our method is available in the GitHub[1].

[1] https://github.com/qianliu0708/PivotsCR

5.2 Background

In this section, we introduce representative candidate retrieval methods for XEL.

Lexicon-based methods. For the monolingual candidate retrieval task, candidate retrieval mainly relies on string matching or mention-entity lexicons [133–136]. For a cross-lingual entity linking task, Wikipedia inter-language resources are employed to fill the cross-lingual gap, such as parallel Wikipedia titles, inter-language entity links. Several lexicon-based candidate retrieval methods have been widely-used in existing state-of-the-art XEL systems [137–139]. For example, Tsai and Roth [25] build a direct probabilistic mapping table using parallel Wikipedia titles and the anchor text mappings, between the source-language and English. It first extracts a source-language mention-entity map from anchor-text mapping in Wikipedia pages. Then, the source-language entity is redirected to its corresponding English entity using the Wikipedia inter-language links. Pan et al. [34] and Zhang et al. [131] proposed to induce word-by-word translations using parallel Wikipedia titles, and used the translated mention to retrieve candidate entities from an existing English mention-entity map. This improved method reduces reliance on source-language anchor-text mapping. Lexicon-based methods are effective for high-resource languages, such as Spanish, but they rely heavily on the coverage of Wikipedia resources, resulting in restrictions on low-resource languages.

Semantic-based methods. Word semantic representation methods [8, 117], which encode meanings of words to low dimensional vector spaces, have become very popular in natural language processing and information retrieval, such as query expansion [140] and text classification [17]. Recently, pre-trained multilingual word representations [35, 141, 142] have been employed to bridge the cross-lingual gap. These methods learn a mapping function to align the source and target embedding space, where synonyms of different languages have similar embeddings. The mentions and entities are represented as fixed-length vectors. Candidate entities retrieval can be undertaken by searching the nearest neighbors of each mention in the embedding space. However, a single low-dimensional embedding has limited representation capacity for mentions or entities [132].

Moreover, powerful pre-trained language models (e.g., Multilingual-BERT) have powerful representation capacities, but they are cost-prohibitive for the candidate retrieval step.

Pivoting language methods. These methods improve the performance of candidate retrieval for low-resource languages (LRL) using a closely related high-resource language (HRL) as an intermediate pivot. For example, *Poland* in Marathi and Hindi are written similarly, and Hindi can be used as a pivoting language for Marathi. Rijhwani et al. [26] train a neural character level string matching model to encode the LRL mentions by leveraging HLR training data. Zhou et al. [129] show that the character-level string matching can be further improved with character n-gram information [143] and extending entity-entity pairs with mention-entity pairs in the training process.

Transliteration methods. These methods are employed when the source-language and English word pairs have similar pronunciation. For example, Upadhyay et al. [144] use a sequence-to-sequence model and a bootstrapping method to transliterate low-resource entity mentions using extremely limited training data. Tsai and Roth [145] combine the standard translation method for candidate retrieval with a transliteration score to improve candidate recall.

Different from the previous methods, our method jointly leverages semantic retrieval and lexical retrieval to search candidate entities for source-language mentions. We learn an intermediary collection with several plausible English mentions to fill the cross-lingual gap and mention-entity gap. Thus, the effective lexical retrieval model used in English can be adapted to other languages.

5.3 Task Description

Cross-lingual entity linking aims to link mentions in a source language to entities in a knowledge base which is written in a target language. It usually consists of two steps: candidate retrieval and entity disambiguation. In this work, we mainly focus on the candidate retrieval component, which plays a critical role in cross-lingual entity linking. For a better understanding, we elaborate on the terminology and corresponding examples

Table 51: Terminology and the corresponding description and examples used in the cross-lingual candidate retrieval task.

Term	Description	Examples
Source language	the language of the text to be linked to KB	French
Target language	the language of the used structural KB	English
Mention	a piece of text in a sentence/question to be linked to KB	pont de Manhattan
Gold Entity	the correct entity in KB for the mention	Manhattan_Bridge
Candidate Entity	retrieved entity from KB for the mention	Bridges_of_Dee

in Table 51. Formally, given a set of source-language mentions $\mathcal{M} = \{m_1, m_2, \cdots, m_{|\mathcal{M}|}\}$ and a target-language knowledge base **K** which contains millions of entities, the goal of candidate retrieval is to retrieve a list of candidate entities $\mathcal{E}_i = \{e_{i1}, e_{i2}, \cdots, e_{iN}\}$ from **K** for each mention $m_i \in \mathcal{M}$, where N is the size of each candidate list.

As the final results of XEL are only generated from candidate entities in $\mathcal{E}$, the candidate list should be as comprehensive as possible to ensure that gold entities are included. Therefore, candidate retrieval methods are measured by *recall*, which is the percentage of retrieved candidate lists that contain corresponding gold entities. Suppose the gold entity of mention m_i is $\hat{e}_i$, *Recall@N* is defined as,

$$Recall@N = \frac{\sum_{i=1}^{|\mathcal{M}|} I(\hat{e}_i \in \mathcal{E}_i)}{|\mathcal{M}|}, \tag{5.1}$$

where $I(\cdot)$ is the indicator function which is set to 1 if true else 0, $|\mathcal{M}|$ is the number of mentions, and N is the number of candidate entities in the retrieved list $\mathcal{E}_i$.

5.4 Pivot-based Candidate Retrieval

To bridge these two gaps, the key idea of our method is to learn an intermediary collection of target-language words which are semantically similar to the source language mention and lexically similar to the target-language gold entity. Figure 52 illustrates our method. The proposed method consists of three stages.

- First, we generate an initial intermediary collection of target-language words using cross-lingual semantic representations. It fills the cross-lingual gap and does not

rely on Wikipedia bilingual resources [34, 138], such as anchor-text links and inter-language links. In addition, high-quality and publicly available multilingual word representations, such as MUSE [35], have a better ability than bilingual lexicons to find a comprehensive collection of related words.

- Second, we design a selective mechanism to refine the initial intermediary collection. The goal is to alleviate the duplication and coverage issue, and thus empower the following lexical search to retrieve a more comprehensive set of candidates.

- Third, we fill the mention-entity gap using lexical retrieval. Each mention is represented as target-language string queries based on the intermediary collection, and the lexical retrieval model uses string overlap information to score mention-entity pairs.

The main contribution of this work lies in the framework which effectively combines the advantage of semantic-based and lexical-based retrieval, and a flexible selective mechanism in the framework which can alleviate the duplication and coverage issues.

For the sake of convenience, we assume the source language is Spanish and the target language is English to illustrate our method in the following section.

5.4.1 Filling the Cross-lingual Gap

Given a Spanish mention $m = \{x_1, x_2, \cdots, x_k\}$ which contains k words, we first generate a set of English words as the intermediary collection $\mathscr{P}$, by searching the English vocabulary. The collection $\mathscr{P}$ aims to represent the semantics of m as comprehensively and accurately as possible to bridge the gap between source and target languages.

Inspired by Lample et al. [35], we employ bilingual word-by-word induction with the help of cross-lingual word embeddings. This process involves (1) aligning source and target embedding spaces and (2) retrieving English words for each Spanish word x_i in m.

Let $\mathscr{X}$ and $\mathscr{Y}$ be the Spanish and English embedding spaces[2], respectively. We learn a

[2]In our method and experiments, we employ the fastText to train monolingual word embeddings: `https://fasttext.cc/docs/en/crawl-vectors.html`

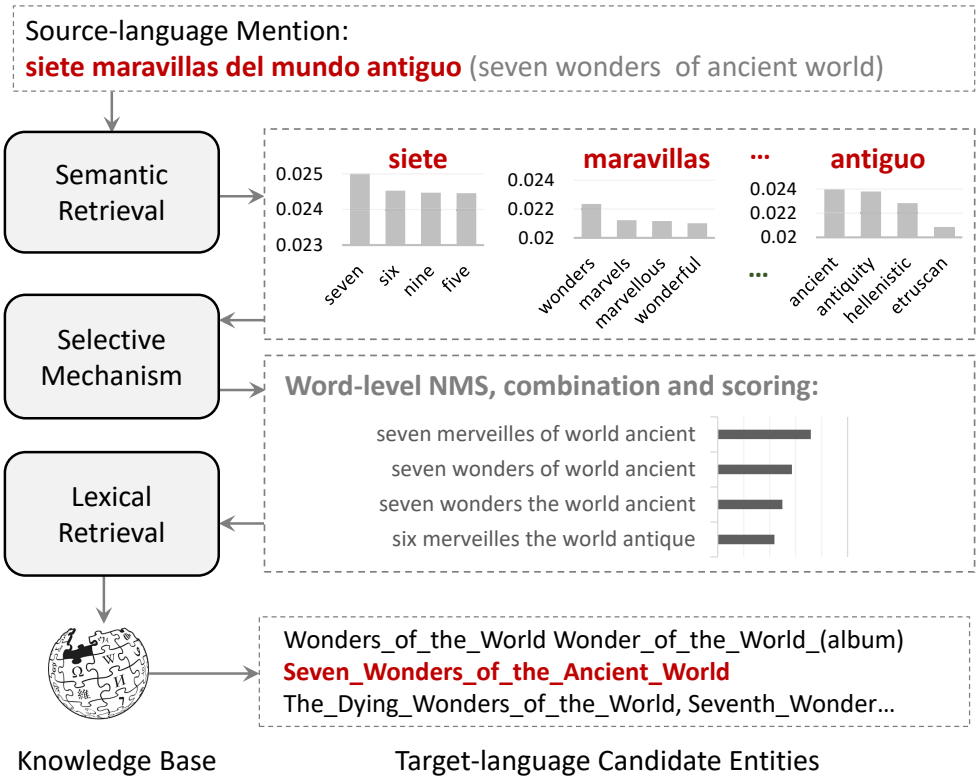


Figure 52: An example to illustrate our pivot-based approach.

mapping $\mathbf{W} \in \mathbb{R}^{d \times d}$ from $\mathcal{X}$ to $\mathcal{Y}$ to align the two spaces, with the objective that synonyms have similar representations. Concretely, we use a seed dictionary of l pairs of words $\{x_i, y_i\}_{i \in \{1,l\}}$, and learn the linear mapping by optimizing,

$$\mathbf{W}^* = \arg\min_{\mathbf{W} \in \mathbb{R}^{d \times d}} \|\mathbf{W}\mathbf{X} - \mathbf{Y}\|_{\mathrm{F}}, \tag{5.2}$$

where d is the dimension of the embeddings, $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{d \times l}$ are corresponding word embeddings of word pairs in the seed dictionary, and $\|\cdot\|_{\mathrm{F}}$ indicates the Frobenius norm. To improve the performance, following Xing et al. [146], we impose an orthogonality constraint on $\mathbf{W}$, i.e., $\mathbf{W}\mathbf{W}^\top = \mathbf{W}^\top\mathbf{W} = \mathbf{I}$. The optimization of $\mathbf{W}$ corresponds to the singular value decomposition (SVD) of $\mathbf{Y}\mathbf{X}^\top$,

$$\begin{aligned} \mathbf{W}^* &= \mathbf{U}\mathbf{V}^\top, \\ \text{with } \mathbf{U}\Sigma\mathbf{V}^\top &= \mathbf{SVD}\,(\mathbf{Y}\mathbf{X}^\top). \end{aligned} \tag{5.3}$$

Then, we retrieve English words for each Spanish x_i in mention m. Specially, x_i is represented by applying projection matrix $\mathbf{W}$ on its Spanish embedding $\mathbf{x}_i$, as $\mathbf{x}_i^* = \mathbf{W}\mathbf{x}_i$.

Next, we explore the nearest English words to $\mathbf{x}_i^*$ in $\mathcal{Y}$. To measure the similarity between Spanish word x_i and each English word y we use the cross-domain similarity local scaling metric (CSLS),

$$\text{CSLS}(\mathbf{W}\mathbf{x}_i, \mathbf{y}) = 2\cos(\mathbf{W}\mathbf{x}_i, \mathbf{y}) - r_{\mathcal{Y}}(\mathbf{W}\mathbf{x}_i) - r_{\mathcal{X}}(\mathbf{y}). \tag{5.4}$$

Here $\mathbf{y} \in \mathbb{R}^d$ denotes the embedding of word y in $\mathcal{Y}$. $r_{\mathcal{Y}}(\mathbf{W}\mathbf{x}_i)$ is the mean similarity of $\mathbf{x}_i$ to its K neighborhoods in $\mathcal{Y}$,

$$r_{\mathcal{Y}}(\mathbf{W}\mathbf{x}_i) = \frac{1}{K} \sum_{y' \in \mathcal{N}_{\mathcal{Y}}(\mathbf{W}\mathbf{x}_i)} \cos(\mathbf{W}\mathbf{x}_i, \mathbf{y}'), \tag{5.5}$$

where $\cos(\cdot)$ denotes the cosine similarity, $\mathcal{N}_{\mathcal{Y}}(\mathbf{W}\mathbf{x}_i)$ is the K neighborhoods associated with $\mathbf{W}\mathbf{x}_i$ in $\mathcal{Y}$. Similarly, $r_{\mathcal{X}}(\mathbf{y})$ denotes the mean similarity of a target word y to its neighborhoods. We refer readers to Johnson et al. [147] and Lample et al. [35] for more details. We employ CSLS here because it significantly increases the accuracy of word retrieval and does not require any parameter tuning.

We select K English words for each Spanish word x_i in mention m, and combine them as the intermediary collection, i.e., $\mathcal{P}(m) = \{y_{1,1}, y_{1,2}, \cdots, y_{1,K}, \cdots, y_{k,1}, y_{k,2}, \cdots, y_{k,K}\}$. Each English word $y_{i,j} \in \mathcal{P}(m)$ is assigned with a score, i.e., $\text{CSLS}(\mathbf{W}\mathbf{x_i}, \mathbf{y_{i,j}})$.

Moreover, in order to alleviate the out-of-vocabulary problem for Spanish word embedding, we also employ multilingual character embeddings [26] to estimate the similarity between x_i and each English word y_j, and retrieve x_i's K most similar English words. We detail the multilingual character embedding training and retrieval, and evaluate its effectiveness in Section 5.5.4.1.

Compared to the lexicon-based approach, our method only relies on a small bilingual dictionary (around 5K word pairs) to align the source and target embedding spaces.

5.4.2 Selective Mechanism

The initial intermediary collection $\mathcal{P}$ suffers from duplication and coverage issues in its role to connect the Spanish mention and English candidate entities. For example, the top-5 retrieved English words for the Spanish word *maravillas* (*wonders* in English) are

{*miracle*, *miracles*, *miraculous*, *miraculously*, *wonderful*}. The duplication issue arises because multiple words have the same meaning with different morphologies, leading to a large number of the same candidate entities appearing repeatedly in the downstream retrieval. The coverage issue arises because some important words with lower similarity are ignored, e.g., the word *wonders* is excluded in $\mathscr{P}(marvillas)$. The low diversity of intermediate sets may result in incomplete candidate entities.

To alleviate these issues, we employ a selective mechanism to refine the intermediary collection. Inspired by the non-maximum suppression (NMS) algorithm [148] that is used to prune redundant bounding boxes in object detection [149] and candidate answer spans in machine reading comprehension [150], we design a word-level NMS to prune morphological variations and improve diversity. Given the initial intermediary collection $\mathscr{P}(x_i) = \{y_1, y_2, \cdots, y_k\}$, after selecting the word y_a which possesses the maximum score, we remove it from the set $\mathscr{P}(x_i)$ and add it to $\mathscr{P}_{NMS}(x_i)$, and delete any y_n in $\mathscr{P}(x_i)$ that is a duplication to y_b. We define that two words are duplicates of each other if they are the same after stemming. This process is repeated for the remaining words in $\mathscr{P}(x_i)$, until $\mathscr{P}(x_i)$ is empty or the size of $\mathscr{P}_{NMS}(x_i)$ reaches a maximum threshold T_w. The time complexity of NMS method is $O(N)$, which is an efficient method to refine the intermediary collection. Algorithm 3 details the word-level NMS method.

Next, we use the softmax function to normalize the word scores in $\mathscr{P}_{NMS}(x_i) = \{y_1, \cdots, y_{T_w}\}$. For mention $m = \{x_1, x_2, \cdots, x_k\}$ with k words, we generate all T_w^k combinations[3]. We denote these combinations as *plausible English mentions* because they may be out of word order. For each plausible English mention we denote its relevance score to the original Spanish mention m as the averaged score of words in it, and T_m plausible English mentions with the highest scores are selected in the final intermediary collection $\mathscr{P}(m)$.

Equipped with the selective mechanism, semantic retrieval is capable of generating diverse English words which are related to the original Spanish word, and avoids the vocabulary mismatch problem from which bilingual lexicon-based methods suffer.

[3]Note we set $T_w = 10$ and usually $k <= 2$, so there are only about 100 combinations. So the time cost for this step is very small.

Algorithm 3 Word-level NMS

Input: $\mathscr{P}(x_i) = \{y_1, \cdots, y_k\}$; $\mathscr{S}(x_i) = \{s_1, \cdots, s_k\}$; T_w
$\mathscr{P}(x_i)$ is the set of candidate translations
$\mathscr{S}(x_i)$ is the set of corresponding scores for word in $\mathscr{P}(x_i)$
T_w denotes the maximum size threshold

Initialize $\mathscr{P}_{NMS}(x_i) = \{\}$
while $\mathscr{P}(x_i) \neq \{\}$ and len($\mathscr{P}_{NMS}(x_i)$)$\leq T_w$ **do**
  $s_a = \arg\max \mathscr{S}$
  $\mathscr{P}_{NMS}(x_i) = \mathscr{P}_{NMS}(x_i) \cup \{y_a\}$
  $\mathscr{P}(x_i) = \mathscr{P}(x_i) - \{y_a\}$
  $\mathscr{S}(x_i) = \mathscr{S}(x_i) - \{s_a\}$
  for $y_b \in \mathscr{P}(x_i)$ **do**
    if stem(y_a)==stem(y_b) **then**
      $\mathscr{P}(x_i) = \mathscr{P}(x_i) - \{y_b\}$; $\mathscr{S}(x_i) = \mathscr{S}(x_i) - \{s_b\}$
    end if
  end for
end while
Return $\mathscr{P}_{NMS}(x_i)$

5.4.3 Filling the Mention-Entity Gap

Given the final intermediary collection of plausible English mentions, we search the candidate entities from the knowledge base using each element of the collection.

We first construct a search space with all the entities in the knowledge base. Each entity is represented by splitting its surface string into words and converted to lowercase. For example, `Manhattan_Bridge` is converted to *manhattan bridge*, `ChessPlayer` is converted to *chess player*. The lexical retrieval model uses word overlap information to score query-entity pairs. We use BM25 [151] to generate the query-entity score based on query statistics and entity statistics. The lexical matching score of a plausible English mention q and an entity e is defined as,

$$lex\_score(q,e) = Sim(q,m) \cdot \text{BM25}(q,e), \tag{5.6}$$

where $Sim(q,m)$ is the relevance score of plausible English mention q to its original Spanish mention m. The top N entities are selected as the candidate entities according to their lexical score.

In the process of bridging mention-entity gap, our method is flexible compared with hard matching methods using anchor-text links. It also runs quickly to search the whole entity space because statistics-based lexical retrieval is more efficient than the high dimensional vector retrieval used in semantic-based methods.

5.5 Experiments

5.5.1 Datasets

We evaluate our method on the following two cross-lingual entity linking datasets, spanning 11 languages.

- **QALD**: We collect cross-lingual entity linking data from the multilingual QALD dataset[4], which is a benchmark for the task of cross-lingual question answering over knowledge base (KBQA). The first step of KBQA is XEL, which links *mentions* in other languages to their corresponding entities in the English KB. Each item in this dataset contains a *question*, *mentions* in this question, and the *SPARQL* to answer this question. We extract gold entities of mentions from the *SPARQL* query. The used knowledge base is DBpedia[5], with 6 million entities. Specifically, we merge all multilingual QALD data, from QALD-4 to QALD-9, and filter out questions whose SPARQL cannot be executed in this knowledge base. For the remaining data, we collect all mentions and their corresponding gold entities to perform the candidate retrieval task. These mentions are from eight languages, namely German, French, Russian, Spanish, Italian, Dutch, Romanian, and Portuguese. We released the used QALD data in our experiment on Github[6].

- **WIKI-LRL**: This is a cross-lingual entity linking dataset[7] for low-resource languages (LRL) collected by Zhou et al. [129]. The knowledge is Wikipedia. The

[4]The dataset is available on `https://github.com/ag-sc/QALD`.
[5]We use the DBpedia 16-10 version: `https://wiki.dbpedia.org/downloads-2016-10`
[6]`https://github.com/qianliu0708/PivotsCR/tree/main/QALD_data`
[7]This dataset is available in `https://github.com/shuyanzhou/pbel_plus`.

Table 52: Top-1000 recall (R@1000) of different methods on the QALD dataset. #Mentions denotes the number of mentions for each language in QALD.

Languages (#mentions)	**German** (672)	**French** (672)	**Russian** (309)	**Spanish** (621)	**Italian** (672)	**Dutch** (621)	**Romanian** (615)	**Portuguese** (309)	**Average**
TRANS-Match	0.525	0.365	0.375	0.422	0.451	0.514	0.514	0.434	0.450
TRANS-Search	0.609	0.588	0.458	0.562	0.570	0.607	0.486	0.553	0.554
SemSearch	0.579	0.484	0.518	0.507	0.540	0.452	0.512	0.489	0.510
Spotlight	0.430	0.342	0.346	0.396	0.374	0.443	-	0.469	0.400
TagMe	0.338	-	-	0.316	-	-	-	-	0.327
OurMethod	0.824	0.801	0.722	0.815	0.799	0.828	0.828	0.812	0.804

candidate retrieval is conducted on 2 million entities of proper nouns in Wikipedia. The mentions are in three low-resource languages, namely Marathi (Indo-Aryan language spoken in Western India, written in Devanagari script), Lao (a Kra-Dai language written in Lao script), and Telugu (a Dravidian language spoken in southeastern India written in Telugu script).

In our experiments, we compare our methods with other candidate retrieval methods on these two challenging datasets. Previous works [152] show that most of the candidate retrieval methods perform well on the Wikipedia-based dataset but fail to generalize beyond Wikipedia, to news and social media text. For a more convincing evaluation, we collect the QALD dataset where mentions are extracted from the user's short search question. Moreover, the existing low-resource XEL performance still lags far behind its high-resource counterparts [129]. We use the low-resource WIKI-LRL dataset to evaluate the robustness of our method to low-resource scenarios.

5.5.2 Baselines

We compare our method with the following five candidate retrieval methods, including lexicon-based methods and semantic-based methods.

- **TRANS** [34]: This is the most widely used lexicon-based candidate retrieval method for state-of-the-art XEL systems such as XELMS [139]. It translates the source-language mention into English in order to predict the entity link. Following Rijhwani et al. [26], we generate a bilingual lexicon with word alignments on parallel

Wikipedia titles[8] using `fast_align` [153], which is a fast and unsupervised word aligner. Each word in the source-language mention is translated into English words using the lexicon. Then we experiment with two varieties to generate candidate entities. **Match** employs the English mention-entity lookup table[9] to generate candidate entities. **Search** utilizes the translated mention as a query and generates candidate entities by a lexical search of the entity space.

- **SemMatch** [35]: This is a semantic-based candidate retrieval method, leveraging cross-lingual word embeddings [141]. Following Pan et al. [154], we convert source-language mentions and target-language entities as fixed-length vectors in an aligned embedding space. We use the approximate nearest neighbors search tool to generate candidate entities. We use MUSE[10] to learn the aligned multilingual word embeddings. Each mention and entity are represented as averaged vector of words it contains. It is notable that some aggregation methods (such as BiLSTM and Transformer) are more powerful, however they are too complex for large-scale entity representation and retrieval to be feasible.

- **Spotlight** [155]: This is a publicly available tool[11] to automatically annotate mentions of DBpedia resources in text, providing a solution for linking unstructured information sources to the structural DBpedia. In our experiment, we use the pyspotlight[12], which is a thin python wrapper around the DBperdia Spotlight and supports ten languages including German, Dutch, French, Italian, and Spanish.

- **TagMe** [156]: This is a fast tool[13] to efficiently and judiciously augment plain text with the corresponding entities in Wikipedia. It is available in English, German and in Italian. We use the tagme-python version[14] in our experiment.

[8] The parallel Wikipedia titles are available in `https://linguatools.org/tools/corpora/wikipedia-parallel-titles-corpora/`.
[9] `https://github.com/dbpedia/lookup`
[10] `https://github.com/facebookresearch/MUSE`
[11] `https://www.dbpedia-spotlight.org/`
[12] `https://github.com/ubergrape/pyspotlight`
[13] The official TagMe API: `https://tagme.d4science.org/tagme/`.
[14] `https://github.com/marcocor/tagme-python`

- **PBEL** [26]: This is a pivot-based entity linking for low-resource language (LRL) tasks. It performs cross-lingual string matching based on an entity gazetteer between a related high-resource language and English. This method removes reliance on the resource of LRL, and achieves state-of-the-art for candidate retrieval in low-resource XEL. In our experiment, we compare our method with PBEL on the WIKI-LRL dataset.

5.5.3 Main Results

5.5.3.1 Comparison on QALD

We first conduct the evaluation of different candidate retrieval methods on the QALD dataset. Table 52 shows the overall performance of our method as well as the baseline methods on the QALD dataset. The gold entity recall of top-1000 (R@1000) candidate entities is reported. We observe that,

- our method performs the best compared with the baseline methods mainly because it leverages both semantic matching and lexical matching information.

- our method and TRANS-Search both use lexical retrieval to generate candidates from the entity space. Our method significantly outperforms TRANS-Search, which implies that the plausible English mentions generated in our method perform much better than the lexicon generated from parallel Wikipedia titles. This indicates that semantic matching information is helpful in candidate retrieval for XEL. TRANS-Search performs slightly better than TRANS-Match, indicating lexical retrieval is more effective than a lookup table.

- the SemSearch method also employs semantic retrieval to fill the cross-lingual gap. It performs worse than our method mainly because a low-dimensional vector is not so accurate enough to represent a mention or an entity, resulting in an inaccurate mention-entity similarity measure. Our method employs plausible

Table 53: Comparison of different methods in terms of average recall on QALD dataset. CR denotes the candidate retrieval in XEL. ED denotes entity disambiguation on the top-1000 candidate entities.

	Avg.	TRANS-Search	SemSearch	Ours
CR	**R@50**	0.381	0.408	0.544
	R@200	0.436	0.434	0.719
	R@500	0.513	0.467	0.765
	R@1000	0.554	0.510	0.804
ED	**R@1**	0.399	0.356	0.573
	R@5	0.486	0.451	0.739
	R@10	0.502	0.468	0.763

English mentions as pivots, and leverage lexical matching information to improve the accuracy.

- our method achieves better performance than Spotlight and TagMe. This indicates that our method is more flexible and feasible for mentions extracts from a user's actual questions.

For a more comprehensive comparison, we vary the size of the candidate entities in range of $\{50, 200, 500, 1000\}$, and report the average recall of TRANS-Search, SemSearch, and our method in Table 53. Moreover, we take the top-1000 candidate entities as input, and perform downstream entity disambiguation using the state-of-the-art method, i.e., multilingual-BERT [10]. For each mention-entity pair, we concatenate the *question* where the mention extracted from and the short *abstract* of the entity as a string, and perform entity disambiguation as the text classification task. The training data is collected from LC-QuAD [157], which is an English KBQA task. Similar to QALD, we extract questions and their corresponding mentions and entities to train the classifier. Table 53 reports the average recall at the top-1, top-5, and top-10 of different methods in entity disambiguation. We observe that,

- in candidate retrieval (CR), our method is consistently superior to other methods with different sized candidate entities, indicating the robustness of our method, and

Table 54: (R@30) on WIKI-LRL. PBEL_Char and PBEL_BiLSTM denote the PBEL method which encodes entities into vectors using BiLSTM and character-based CNN, respectively.

Languages (#mentions)	**Marathi** (2449)	**Lao** (799)	**Telugu** (1742)	**Average**
SemSearch	0.596	0.195	0.418	0.403
PBEL_BiLSTM	0.535	0.210	0.407	0.407
PBEL_CharCNN	0.477	0.180	0.246	0.348
OurMethod	0.702	0.307	0.532	0.514

- in entity disambiguation, pre-trained language model (i.e., multilingual-BERT) is powerful to learn the relevance between the source-language text and the target-language entity. Compared with the other method, our method achieves better performance, mainly due to the high recall in upstream candidate retrieval.

5.5.3.2 Comparison on WIKI-LRL

Then, we compare our method with the other baselines on the WIKI-LRL dataset in Table 54. Following [158], we report top-30 gold candidate recall. In the WIKI-LRL dataset, the source-language mentions are Wikipedia titles and the TRANS methods that rely directly on the Wikipedia titles as lexicons are excluded from the comparison. We observe that our method achieves the best performance across all three languages. PBEL is the state-of-the-art candidate retrieval method for low-resource language, and it is effective to leverage related high-resource languages as pivots to reduce the disconnect between mentions and entities. Our method leverages plausible English mentions as an intermediate without additional high-resource language information and achieves better results. Compared with SemSearch, our method performs better mainly because it combines the semantic similarity and lexical similarity between the mention and entity using plausible English mentions as the intermediary, instead of directly computing their similarity in the aligned latent space.

Table 55: R@1000 on the QALD dataset to investigate the influence of character information. OOV denotes the percentage of our-of-vocabulary mentions. Δ denotes the performance improvement.

Languages	**OOV(%)**	**w/ Char**	**w/o Char**	Δ
German	4.17%	0.821	0.824	0.003
French	4.03%	0.796	0.801	0.004
Russian	4.17%	0.718	0.722	0.003
Spanish	4.17%	0.805	0.815	0.010
Italian	4.03%	0.786	0.799	0.013
Dutch	3.74%	0.821	0.828	0.006
Romanian	3.88%	0.811	0.828	0.016
Portuguese	3.88%	0.780	0.812	0.032
Average	4.01%	0.792	0.804	0.012

5.5.4 In-depth Analysis

The intermediary collection $\mathscr{P}$ plays an important role in our method. To analyze the performance of different modules and investigate their impact on the final results, we evaluate the effect of character information, word-level NMS, and the size of the intermediary collection. Then, we analyse the bilingual-resource reliance and time-efficiency of our method.

5.5.4.1 Effect of Character Information

When filling the cross-lingual gap, if a source-language word x_i is out of vocabulary of the embedding space $\mathscr{X}$, we cannot find its semantically related English words. Inspired by the previous method [26, 143], we leverage character-level semantic matching to alleviate this problem.

To be specific, we randomly initialize all the characters in the source and target languages as fixed-length embeddings. Then, we design two character-level BiLSTM to encode words in the source and target languages in the latent vector space. Consider a source-language word x_i and its parallel target-language word y_i. Each word is a sequence of characters. The character embeddings are used as input to the BiLSTM and the final character embedding of each word is the concatenation of the last states

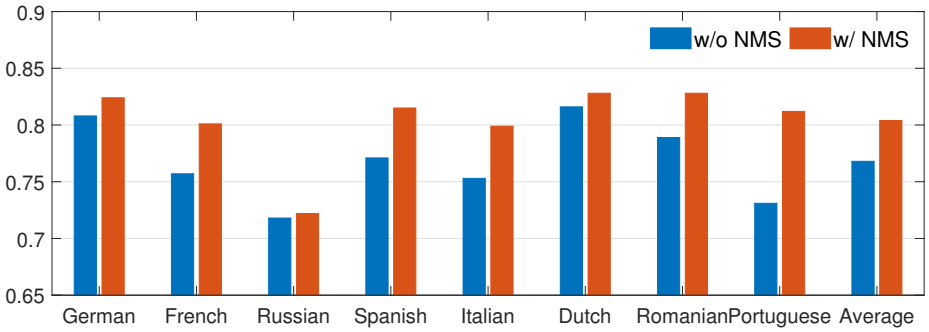


Figure 53: R@1000 on QALD to investigate the effectiveness of the NMS component.

from the forward and backward LSTMs. We train the model with a max-margin loss to maximize the cosine similarity between words which have same meaning in different languages, and minimize the similarity between negatively sampled word pairs:

$$\mathscr{L} = \max(0, sim(x, y^{-}) - sim(x, y) + \lambda), \tag{5.7}$$

where x and y is a word-pair in the seed dictionary which have the same meaning, y^{-} is a negative word in target language, and λ is the margin.

In our experiment, for the out-of-vocabulary source-language words, we search its most similar target-language words according to their character cosine similarities. We evaluate the performance of character information in the QALD dataset in Table 55. We observe that 4% mentions are out-of-vocabulary in word-level embedding space. Character-level information helps to improve our method, with an average performance gain of 1.2%.

5.5.4.2 Effect of Word-level NMS

We assess whether the word-level NMS is effective for generating diverse English mentions in Figure 53. We observe that our method achieves a significant performance gain using the word-level NMS method, with an averaged performance gain of 3.6%. This improvement is mainly comes from duplication reduction of the NMS component, which enhances the diversity of the intermediary collection and better covers the salient information in the mention.

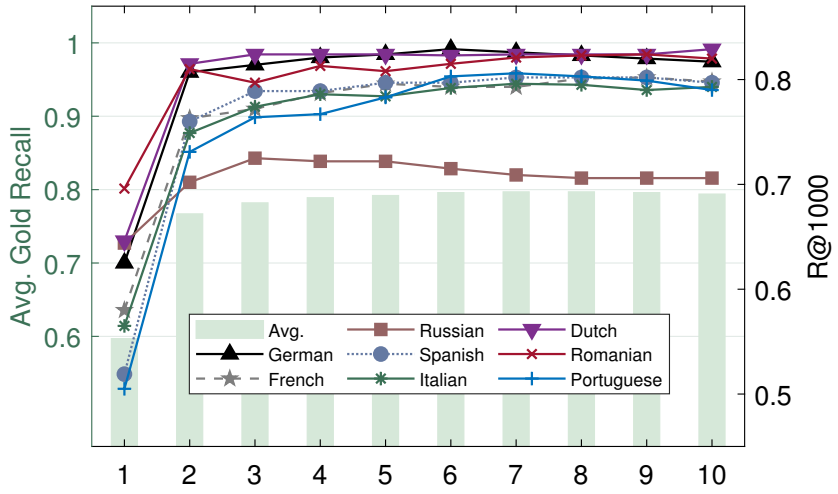


Figure 54: Influence of the size of intermediary collection on the QALD dataset. The x-axis shows the size of the intermediary collection, the left y-axis corresponds to the average R@1000 across eight languages, and the right y-axis denotes R@1000 of each language.

5.5.4.3 Size of the Intermediary Collection

For each source-language mention m, we generate an intermediary collection with T_m plausible English mentions. To investigate the influence of T_m on candidate retrieval, we vary T_m between 1 and 10. The detailed results of R@1000 for different languages are plotted in Figure 54. The green bars represent the averaged recall of different languages. We observe that it performs worst when P_m only contains one plausible English mention (i.e., $T_m = 1$). This is mainly because that a word or phrase usually has multiple expressions, and one plausible English mention may be inaccurate and incomplete to capture the original source-language mention. Our method achieves best performance when T_m is set to 7. It is notable that adding plausible English mentions will result in a linear increase in time complexity of the lexical retrieval process. In practice, the recommend T_m is in range of [3,7].

5.5.4.4 Bilingual Resource Reliance

Our method only needs a bilingual word dictionary to align the source and target embedding space, which is a low-resource reliance method. We compare our method with

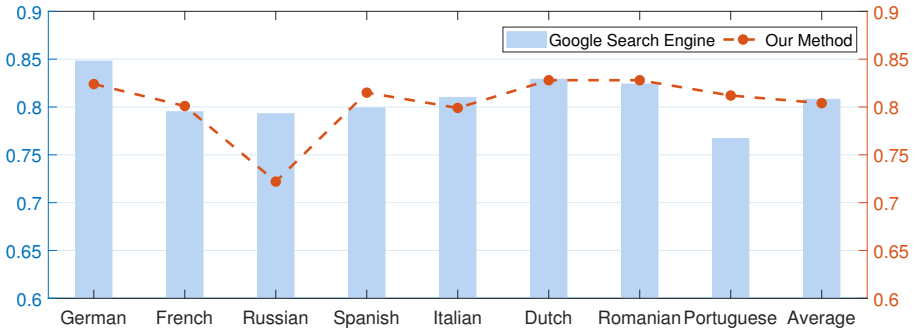


Figure 55: Comparison between our method and Google translator on the QALD dataset. The y-axis denotes R@1000 score of candidate retrieval.

the Google translator, which translates source-language mentions to target-language mentions and then generates candidate entities with lexical retrieval using BM25. It is important to note that the Google translator is trained on massively bilingual resources and is not available in many practical and industry scenarios. Figure 55 compares the performance of our method with the Google translator on the QALD dataset. The blue bar denotes the performance of Google translator in different language. The red line denotes the performance of our method. We observe that our method achieved better performance for Portuguese, Spanish, and French, but a bit worse for Russian. Considering the average R@1000 of eight languages, the Google translator (i.e., 0.808) only achieves a slight improvement of 0.4% over our method (i.e., 0.804). This demonstrates the effectiveness of intermediate collection, and the effectiveness of semantic retrieval and selection mechanisms in filling the cross-language gap.

5.5.5 Case Study

In this section, we present several examples from the QALD dataset in Figure 56 to give an intuitive impression of our method. We present the source mentions, their corresponding gold entities, and plausible English mentions generated by our method. We observe the plausible English words are effective to fill the cross-lingual gap between source and target language. For example, semantic retrieval is accurate to connect *Finland* in Russian and English scripts. The plausible English mentions that are important to recall

Source Mention	Gold Entity	Plausible English Mentions
Norte Mar (Portuguese)	North_Sea	norte sea, south sea, **north sea**, south sea, southern mar
francés quinto República (Spansih)	French_Fifth_Republic	**french fifth republic**, france five republic, france fourth republic, french fifth republic, french fifth republican
burro di noccioline (Italian)	Peanut_butter	**butter di peanuts**, lard di peanuts, burro di peanuts, butter di custard, burro di syru
Финляндия (Russian)	Finland	**finland**, finnish, sweden, estonia, norway

Figure 56: Examples in the QALD dataset. The red plausible mentions are salient mentions to recall gold entity, marked by human evaluation.

the gold entity in downstream lexical retrieval are marked in red. For example, *butter di peanuts* is an effective query to search the entity `Peanut_butter`.

5.6 Summary

In this chapter, we proposed a pivot-based candidate retrieval method for cross-lingual entity linking. The proposed method leverage multilingual word representations to learn a pivot set, so that the efficient English retrieval model can be applied to other languages. It takes an intermediary set of plausible target-language mentions as pivots to bridge two types of gaps: cross-lingual gap and mention-entity gap. The learned plausible target-language mentions are capable of capturing the semantics of source-language mentions, and are effective to recall gold entity in the lexical retrieval. In the experiments, we evaluate our method on two challenging XEL datasets and the results demonstrate the competitiveness of our method.

CHAPTER 6

FUZZY SIMILARITY MEASURE BASED ON REFINED SEMANTIC REPRESENTATION

6.1 Introduction

Distributed word representation is widely used in text-related tasks. Especially in large-scale retrieval and similarity calculation, word vector has the advantage of fast and low resource consumption compared with large-scale pre-trained language models. In this chapter, our goal is to refine the general word representation based on task-specific characteristics to better support the downstream top-k selection task.

Top-k words selection is the process of selecting the k-most relevant words to a given word from a set of alternatives. The demand for top-k words is a long-standing research problem in many real-world applications, including word sense disambiguation [159] and query expansion [160]. The natural approach for this process is to apply a word similarity measure that compares two words, to return the level of similarity between them. Several methods exist, and they can be roughly classified into two groups: knowledge-based methods and corpus-based methods. Knowledge-based methods rely

on manually-compiled lexicons to measure the similarity between two words, such as WordNet [13]. In practice, these methods are labor-intensive and inflexible because the meaning of many words changes in different contexts and domains, and may also change over time [161]. Corpus-based methods measure the similarity between words using statistical information from the corpus, such as point-wise mutual information [162] or latent semantic analysis [163]. The basic idea is that two words are similar to each other if they frequently co-occur [164].

A promising direction for top-k selection is the distributed representation method, commonly known as word embedding. In this approach, words are mapped from a vocabulary to low-dimensional vectors, which enables the discovery of latent semantics behind the words. The similarity is then measured according to the similarity of the vectors using a number of techniques, such as cosine similarity or Euclidean distance. Methods based on neural networks, such as Word2Vec [8], GloVe [7], are known to be particularly effective for learning high-quality word embeddings from a large-scale corpus. They are also more computationally efficient than many other solutions.

However, despite their effectiveness, the quality of the similarity measured by word embedding methods, known as embedding similarity, is under debate. The crux of the discussion is that most word embedding methods rely on statistics, to show how often each word occurs within the local context window of another word, which means they mainly capture the *proximity* property between words [28]. Whereas in practice, many valuable associative relationships between words could exist across a longer linguistic distance. Thus, word embedding-based top-k selection is biased – the local co-occurrence of words is emphasized, but the global relevance of words is ignored.

To round out the top-k selection, we propose a refined similarity measure as a complement to word embedding that considers both local and global information. More concretely, we consider the global measure using a technique called *association rules*, which is able to discover frequently occurring words over a much longer distance in a corpus. However, different types of similarity measures are used for local and global information, i.e., real-number embedding similarity for local information and association

rules for global information. Therefore, directly transferring one measure to another has been problematic. The solution proposed in this chapter is a fuzzy word similarity measure that combines both types of measures using a fuzzy logic system and fuzzy rules. The advantage of fuzzy logic is that it provides a flexible and convenient way to transform expert knowledge expressed in natural language into fuzzy rules. Inferring a final similarity score for pairs of words by combining different types of measures is also a relatively straightforward task for this type of framework.

We compare the performance of the proposed top-k selection method with eight state-of-the-art baselines on a query expansion task to show its usefulness. The task setting uses similarity measures to select the k-most appropriate words for a given query, and the performance is evaluated by comparing the document retrieval results. Three widely used information retrieval datasets were tested: TREC-disk 4&5, WT10G, and RCV1. The experiment results show that the fuzzy word similarity measure used in our method significantly outperforms the measures used in other state-of-the-art baselines for top-k selection scenarios.

6.2 Background

The purpose of our work is to incorporate fuzzy theory into a word similarity measure method to overcome the bias inherent in the similarities calculated by current word embedding methods. The method is designed to solve top-k selection problems and as such, top-k selection methods, association rules, and fuzzy logic systems are all relevant to our research. Previous work in these four areas are briefly reviewed in this section.

6.2.1 Top-k Word Selection

The most common method for top-k selection is to use knowledge bases. For example, WordNet [13] is a word database that groups nouns, verbs, adjectives, and adverbs into sets of synonyms (called synsets). Each synset is linked by semantic or lexical relationship, such as hypernym, hyponym, meronym, etc. Several approaches [165, 166]

rest on an edge-counting method to measure the distance between words, in either WordNet or another similar database. The longer the path, the less similar the words. Other methods, such as [167, 168], estimate the similarity between words according to the number of common features in the knowledge base, such as synonyms, definitions, and relationships. While the knowledge-based method is efficient, certain techniques can affect its performance. For instance, common features usually need to be annotated manually, and if a word has two different meanings in two different domains, or the meaning of a word changes over time, the method's performance is reduced.

Corpus-based methods are statistical. For example, the latent semantic analysis method [163] generates a term-document matrix using the SVD (singular value decomposition) method to create vector spaces in which words are represented as vectors. Alternatively, the latent Dirichlet allocation method [37] leverages the distributions of words in large collections of documents. The documents are modeled as topic distributions, and the topics are modeled as word distributions given a vocabulary. The point-wise mutual information method [162, 169] uses information retrieval engines to gather co-occurrence statistics based on computed similarity scores.

However, more recently, word embedding methods have taken over as the dominant corpus-based method for measuring word similarity.

6.2.2 Association Rules

Association rules discover long-distance related patterns and have been shown to be useful in selecting rational words, particularly for query expansion tasks [170]. However, applying association rules in query expansion is far from a trivial task, mostly because of the huge number of association rules that can be drawn from a document collection. To alleviate this problem, Latiri et al. [171] proposed a minimal generic basis for retaining only the essential association rules. Bouziri et al. [172] collated a minimal set of rules, allowing for an effective selection of rules to be used in the expansion process. Bouziri et al. [173] proposed a pairwise learning method to rank candidate association rules by selecting the most relevant rules to generate relevant expansion terms. Abbache et al.

[174] reviewed query expansion using Arabic WordNet and association rules. In this work, we use association rules as complementary resources for word embeddings and emphasize its ability to discover word relationships with unlimited distance.

6.2.3 Fuzzy System and Its Application in Similarity Measures

Fuzzy systems are composed of three successive modules, namely fuzzification, inference, and defuzzification. This arrangement can be regarded as a knowledge-based nonlinear system. Fuzzy systems have been used extensively in many applications, including recommendation systems [175], domain adaptation [176, 177], concept drift [178, 179], event extraction [180], multiple periodic factor prediction [181], and image processing [182, 183]. Several studies have clearly demonstrated the advantages of incorporating fuzzy logic into text mining applications. For example, the fuzzy bag-of-word method [184] addresses sparsity and a lack of high-level semantic representations with the BoW method for document classification tasks. Lee and Jiang [185] developed a fuzzy-based method, called ML-FRC, to address multi-class text categorization problems. Here, a fuzzy transformation method is used to construct low-dimensional fuzzy relevance vectors, then the fuzzy clustering is linearly mapped to labels. Martin et al. [186] combined a fuzzy approach based on grammar with incremental learning, where fuzzy grammar fragments extract structured grammar components from unstructured text.

Moreover, much research has shown the benefits of fuzzy methods for top-k selection. Fuzzy algorithm for similarity testing [187] is an ontology-based similarity measure that uses concepts from fuzzy logic and computing with words to generate accurate representations of fuzzy-style words. Singh et al. [188] combine the different weights of each term and fuzzy rules to infer the weights of additional query terms. Singh and Sharan [189] integrate a crisp relevance score for query expansion terms into a fuzzy information system. The query expansion terms are derived from six different expansion term selection methods. They also designed special fuzzy rules to infer the specific weight of each additional term. Liu et al. [190] developed fuzzy rules to re-weigh the expansion terms generated from word embeddings by considering the small variances between

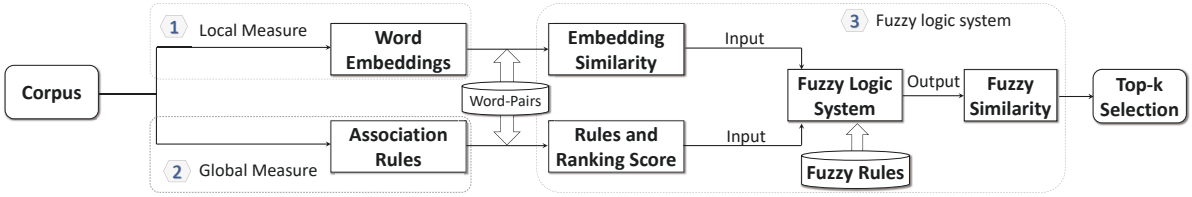


Figure 61: The framework of the proposed fuzzy word similarity measure based on both local measure and global measure in the corpus.

expansion terms. Gupta and Saini [191] modified the query expansion approach with a term weighting method, by employing particle swarm optimization to infer the weights, and fuzzy logic to ensure the optimization is adaptive.

Our method is derived from fuzzy logic systems that integrate crisp similarity measures. Our advancement is to combine different complementary but heterogeneous measures, to refine word embedding-based top-k selection.

6.3 Fuzzy Word Similarity Measure

The framework of our fuzzy word similarity measure, which jointly considers local and global measures, is illustrated in Figure 61. There are three components in our method: (1) local measure, (2) global measure, and (3) fuzzy word similarity computation using a fuzzy logic system. This section details each component.

6.3.1 Local Measure: Word Embedding

Word embedding methods represent each word w in vocabulary $\mathcal{V}$ as a d-dimensional vector $\mathbf{w} \in \mathbb{R}^d$. Of the various word embedding methods, we follow the Word2Vec method proposed in Mikolov et al. [8], which is shown to be a robust baseline by Levy et al. [192].

The Word2Vec method uses extremely computationally efficient log-linear models to produce high-quality word embeddings. A sliding window moves across the corpus, where the central word is the target word, and the other words form the context. Word2Vec is comprised of two models, CBOW and Skip-gram. The CBOW model uses the average or sum of the context words as input to predict the target word, and the Skip-gram model

uses the target word as input to predict each contextual word. Suppose the context-window is $\{w_{i-k},...,w_{i-1},w_i,w_{i+1},...,w_{i+k}\}$, the target word is w_i, and the others are contextual words. The CBOW model aims to predict the target word using contextual words with the following:

$$\mathscr{L}_{cbow} = \arg\max \sum_{i=1}^{N} \log Pr(\mathbf{w_i} \mid \mathbf{w_{context}}), \tag{6.1}$$

where $\mathbf{w_i}$ is the vector of target word w_i, $\mathbf{w_{context}}$ is the average vector of all contextual words, $\mathscr{W}$ represents the vocabulary, and N is the number of words in the corpus. The probability is formulated with a softmax function as

$$Pr(\mathbf{w}_i \mid \mathbf{w}_{context}) = \frac{\exp(\mathbf{w_i} \cdot \mathbf{w_{context}})}{\sum_{w \in \mathscr{W}} \exp(\mathbf{w} \cdot \mathbf{w_{context}})}. \tag{6.2}$$

In contrast to CBOW, Skip-gram aims to predict each contextual word when given the target word. Therefore, the objective of Skip-gram is to maximize the log probability

$$\mathscr{L}_{skipgram} = \arg\max \sum_{i=1}^{N} \sum_{-k \le c \le k, c \neq 0} \log Pr(\mathbf{w_{i+c}} \mid \mathbf{w_i}), \tag{6.3}$$

where the probability is also formulated with a softmax function as

$$Pr(\mathbf{w_{i+c}} \mid \mathbf{w_i}) = \frac{\exp(\mathbf{w_{i+c}} \cdot \mathbf{w_i})}{\sum_{w \in \mathscr{W}} \exp(\mathbf{w} \cdot \mathbf{w_i})}. \tag{6.4}$$

The negative sampling method is widely used to optimize these objective functions. After optimization, words in the vocabulary $\mathscr{V}$ are mapped into a low-dimensional, real-valued vector space. Then, the similarity of word-pairs can be computed using the similarity of the vector in the word embedding space according to a metric, e.g., Euclidean distance or cosine similarity.

6.3.2 Global Measure: Association Rules

Association rule mining has been widely used to discover related patterns in the field of data mining. It is a sensible choice for mining complementary global relatedness for word embeddings. Figure 62 illustrates the difference between word embeddings and association rules. Consider the term *programming* as an example. Word embedding

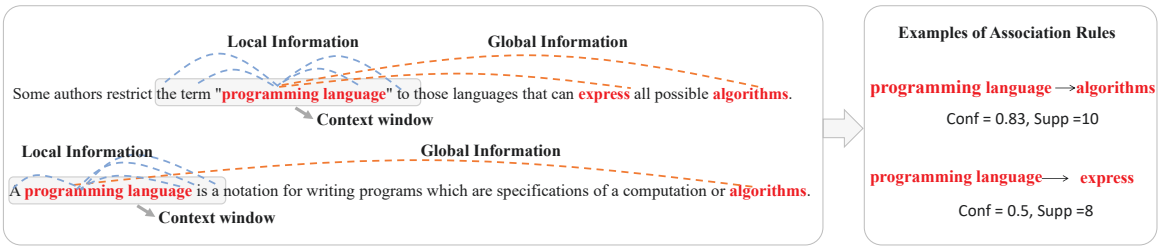


Figure 62: An illustration of local and global information in some example sentences. The context window used in word embedding methods is marked by the pale red rectangles. For *programming*, its local related words and global related words are linked in blue and orange, respectively. Several examples of association rules are listed on the right-hand-side of the figure.

methods only model the co-occurring words within the context window, such as *term*, or *language*. However, these methods ignore other related words, such as *algorithms*, which occur over long distances. Note that by increasing the window size to allow for long distance words, this will introduce more noise information and impair the ability of the word embeddings to capture real related words. Instead of using a large context window, association rule mining can discover valuable relationships among words with the unlimited distance. For example, association rule mining can easily find the relationship between *programming* and *algorithms*, which is beyond the coverage of the context window.

Definition 1 (Association Rule [193])**.** *Let $\mathscr{I}$ be a set of items and $\mathscr{D}$ be a set of transactions, then pattern X and pattern Y construct the association rule $X \rightarrow Y$, if (1) $X \subset \mathscr{I}$, $Y \subset \mathscr{I}$, $X \cap Y = \emptyset$; (2) $Supp(X) \geq T_s$, $Supp(Y) \geq T_s$; (3) $Conf(X \rightarrow Y) \geq T_c$.*

Here, each sentence in the corpus is a transaction in $\mathscr{D}$, and it contains a subset of the items in $\mathscr{I}$. Every rule is composed of two different sets of items, X and Y, where X is denoted as *an antecedent pattern*, and Y is denoted as *a consequent pattern*. *Support*, denoted as $Supp(X)$, is an indication of how frequently X appears in the corpus with respect to $\mathscr{D}$, and is defined as the proportion of transactions t in $\mathscr{D}$ which contains X:

$$Supp(X) = \frac{|t \in \mathscr{D}; X \subseteq t|}{|\mathscr{D}|}. \tag{6.5}$$

Confidence, denoted as $Conf(X \rightarrow Y)$, is an indication of how often the rule is true, which is defined as the proportion of the transactions that contain X which also includes Y:

$$Conf(X \rightarrow Y) = \frac{Supp(X,Y)}{Supp(X)}, \tag{6.6}$$

where $Supp(X,Y)$ is the support that X and Y have occurred together.

In the definition, T_s is a minimum support threshold and T_c is a minimum confidence threshold. The minimum support threshold T_s is applied to find all frequent items in the corpus. Once the frequent items are generated, candidate rules are formed by the binary partition of each itemset. From a list of all candidate rules, the minimum confidence T_c constraint is applied to these frequent itemsets in order to form rules. With a confidence threshold, strongly associated rules are selected.

All selected association rules are collected in $\mathscr{R} = \{r_1, r_2, \cdots, r_n\}$. Each rule is denoted as:

$$r_i = (X_i \rightarrow Y_i, s_i, c_i), \tag{6.7}$$

where X_i is its antecedent pattern, Y_i is its consequent pattern, s_i is the frequency, and c_i is the conditional probability.

6.3.3 Fuzzy Word Similarity Measure

The proposed fuzzy word similarity measure uses a fuzzy logic system to jointly consider the local and the global measures implemented by word embedding and association rules, respectively. Since word embedding and association rules yield heterogeneous similarity measures, the fuzzy logic system is applied here to combine them together. Following the basic framework of fuzzy logic systems, there are three steps in our method: fuzzification, fuzzy logic operators, and defuzzification. Each step is detailed as follows.

6.3.3.1 Fuzzification

This step uses membership functions to quantify the membership degree of all input and output variables. In our method, there are two input variables, i.e., a local score S_{local}

generated from word embeddings and a global score S_{global} generated from association rules. There is only one output variable, i.e., the similarity score S_{fuzzy}.

For w_a and w_b, the local score (S_{local}) input variable represents the similarity between vectors in the word embedding space. Cosine similarity is applied here, i.e.,

$$S_{local} = cos(w_a, w_b) = \frac{\mathbf{w}_a^T \cdot \mathbf{w}_b}{||\mathbf{w}_a|| \cdot ||\mathbf{w}_b||}, \tag{6.8}$$

where $\mathbf{w_a}$ and $\mathbf{w_b}$ denote the word embeddings of two words w_a and w_b, respectively. The cosine similarity measure is high when the words are close to each other, and vice versa.

The other input variable is the global score (S_{global}), which is the similarity derived from the association rules. We designed a ranking mechanism to measure this similarity. The basic idea of the mechanism is to rank all the rules in $\mathscr{R}$ and assign each rule with a ranking score. Then, if two given words co-exist in a rule, its ranking score reflects their global similarity. Most notably, two words may co-exist in many rules, so the maximum ranking score is used as the final global similarity. Specifically, the procedure for computing global similarity is as follows:

Step 1: All rules in $\mathscr{R}$ are sorted according to their frequency. The sorted set of rules is denoted as $\mathscr{R}' = [r^1, r^2, \cdots, r^m]$, in which any two rules $r^i = (X_i \rightarrow Y_i, s_i, c_i)$ and $r^j = (X_j \rightarrow Y_j, s_j, c_j)$ meet the condition that if $i \leq j$ then $s_i \geq s_j$.

Step 2: The weight of r^i is high when i is small, and vice versa. The weight of each rule is directly defined as

$$s(r^i) = (m - i + 1)/m, \tag{6.9}$$

where $1 \leq i \leq m$, and m is the size of $\mathscr{R}$.

Step 3: Rules that contain both words are extracted into a set $\mathscr{R}(w_a, w_b)$. Formally, rule $X \rightarrow Y \in \mathscr{R}(w_a, w_b)$ when 1) $w_a \in X$ and $w_b \in Y$; or 2) $w_a \in Y$ and $w_b \in X$. The maximum ranking score of all the rules in $\mathscr{R}(w_a, w_b)$ is regarded as the global similarity:

$$S_{global}(w_a, w_b) = max\{s(r) | r \in \mathscr{R}(w_a, w_b)\} \tag{6.10}$$

In this way, the two input variables are assigned with crisp scores.

Next, we define the fuzzy set of input and output variables. In our method, each word is represented by a set of fuzzy variables rather than a single one. In the query expansion task, candidate expansion words are usually divided into three categories [2, 194], i.e., *good*, *neutral*, and *bad*. To be specific, given a query, *good* words are related to this query and improve the retrieval effectiveness, *neutral* words are those that produce similar retrieval performance when they are selected into the expansion set, and *bad* words are not related to the query and compromise the effectiveness of retrieval. Accordingly, we represent the fuzzy set of input and output variables as three fuzzy linguistic variables: high (H), medium (M), and low (L), to describe the degree to which a word is related, neutral, and unrelated to the query, respectively.

Then, membership functions are used in the fuzzification and defuzzification steps to quantify a linguistic term and map the non-fuzzy inputs to the defined fuzzy linguistics, or vice versa. The three most common types of membership functions are triangular, trapezoidal, and Gaussian. The selection of the membership function can be context-based, and it is generally chosen according to the specific task and user's experience. In implementation, the Gaussian membership function is applied, where sigma is set to 0.2, and the means for L, M, and H is 0, 0.5, and 1, respectively. In this work, we follow the density distributions of the data concerning the fuzzy variables to select the membership function.

To be specific, we repeat the evaluation of the retrieval model in RCV1 dataset, and in each evaluation the expansion set only consists of one word. The word's *retrieval gain* or *retrieval loss* is calculated by comparing the differences between retrieval precision and the result of the original query [2, 194]. We define *good* words as those with a retrieval gain of more than 0.005 and *bad* words as those with retrieval loss of more than 0.005. The remaining words with a gain or loss smaller than 0.005 are defined as *neutral* words.

Table 61 provides a sense of what is actually generated by local and global measures, i.e., the top-10 words of *crime* and *feeling* according to S_{local} and S_{global} in the RCV1 dataset. Observe that two methods mined several common words. For instance, *mood* and *emotion* were detected as similar words to *feeling* from both the word embeddings and the

Table 61: Examples of the most similar words selected using word embedding and association rules for two given words (*crime* and *feeling*). RCV1 corpus is used here which is detailed in Section IV-A.

crime		feeling	
WordEmbed	AssoRule	WordEmbed	AssoRule
assault	violent	emotion	emotion
gang	rate	frustration	sentiment
mafia	police	sentiment	desire
criminal	money	awful	sensitivity
robbery	fight	mistrust	mood
addiction	federal	mood	sympathy
paedophilia	clinton	frankly	sadness
offense	criminal	pinch	gratitude
homicide	reported	despair	despair
murder	percent	seem	affection

Table 62: The average percentage of the retrieval gain, neutral, and loss, of 50 collections in the RCV1 dataset (detailed in Section IV-A). Common words are detected by both local measure and global measure.

	Local Measure	Global Measure	Common Words
Good	10.8%	14.0%	22.0%
Neutral	76.4%	78.8%	70.0%
Bad	12.8%	7.2%	8.0%

association rules. Table 62 shows the retrieval results of the expansion words generated from the local measure, global measure, and their common words. We observe that most words do not improve the retrieval effectiveness and no more than 15% of the words detected by the local measure and global measure are *good* expansion words. Of the three membership functions, Gaussian membership function tends to detect relatively few words under the same threshold. In our method, a Gaussian membership function is used to quantify the membership degree of all the input and output variables.

6.3.3.2 Fuzzy logic operators

To design the fuzzy logic rules, we evaluate the effectiveness of each expanded word, particularly the common words in Table 62. We observed that:

- Association rules are effective for generating good expansion words.
- Common terms produce a higher percentage of good words than other terms.
- Most words have no significant effect on the retrieval performance, and adding more words increases the risk of introducing noise.

These observations demonstrate the strong potential of using association rules to adapt word embedding-based top-k words selection. However, they also show the need to customize word selection. Based on the above observations, as well as common knowledge of the IR system, we designed nine fuzzy rules.

Firstly, if a word's score-pair (i.e., S_{local} - S_{global}) is high-high, high-medium, or medium-high, it is related to the given query assessed by both local measure and global measure. As such, it is more likely to be regarded as a *good* expansion word with a *High* similarity score. The rules are designed as follows:

Rule 1 : IF $S_{local} = H$ AND $S_{global} = H$, THEN $S_{fuzzy} = H$.

Rule 2 : IF $S_{local} = H$ AND $S_{global} = M$, THEN $S_{fuzzy} = H$.

Rule 3 : IF $S_{local} = M$ AND $S_{global} = H$, THEN $S_{fuzzy} = H$.

Secondly, if a word's score-pair is low-low, medium-low, or low-medium, it is not related to the query either in the local measure or in the global measure. As such, it is more likely to be regarded as a *bad* expansion word with a *Low* similarity score. The rules are designed as follows:

Rule 4 : IF $S_{local} = L$ AND $S_{global} = L$, THEN $S_{fuzzy} = L$.

Rule 5 : IF $S_{local} = M$ AND $S_{global} = L$, THEN $S_{fuzzy} = L$.

Rule 6 : IF $S_{local} = L$ AND $S_{global} = M$, THEN $S_{fuzzy} = L$.

Lastly, other score-pair cases tend to be *neutral* words which do not provide retrieval gain or loss. The rules are designed as follows:

Rule 7 : IF $S_{local} = M$ AND $S_{global} = M$, THEN $S_{fuzzy} = M$.

Rule 8 : IF $S_{local} = L$ AND $S_{global} = H$, THEN $S_{fuzzy} = M$.

Rule 9 : IF $S_{local} = H$ AND $S_{global} = L$, THEN $S_{fuzzy} = M$.

To execute fuzzy rules, the input of each rule is the fuzzy set defined in the fuzzification process, and the output of each rule is a fuzzy set that shows the degree of support for each rule. Since all fuzzy rules should be combined to make a final decision, the output fuzzy set of each rule should be aggregated. We apply the *max* aggregation method to combine all fuzzy sets into a single fuzzy set.

6.3.3.3 Defuzzification

This process defuzzifies the aggregated output fuzzy set, which means quantifying the output as a number to represent the final learned similarity between words. Defuzzification is performed according to the membership function of the output variable. In our method, we use the centroid method [195] for defuzzification, which returns the center of the area under the curve. The algorithm of the proposed fuzzy word similarity measure is shown in Algorithm 4. For the efficiency of our method, it is mainly affected by the progress of mining associated rules.

Algorithm 4 Fuzzy Word Similarity Measure

Input: A corpus $\mathscr{C}$ with a vocabulary $\mathscr{V}$. Settings for training word embeddings, i.e., the dimension of vector d, context window size k, and negative sampling n. Settings for mining association rules, i.e., the threshold of support T_s and confidence T_c. A set of fuzzy rules $\mathscr{F}$ defined in this chapter.

Output: a similarity measure, which can return the similarity between words w_a and w_b ($w_a \in \mathscr{V}$ and $w_b \in \mathscr{V}$).

1: Based on the corpus, train a word embedding matrix $\mathbf{W} \in \mathbb{R}^{|\mathscr{V}| \times d}$ that each row $\mathbf{w} \in \mathbb{R}^d$ is the vector of word w.
2: Based on the corpus, mine association rules under the thresholds T_s and T_c, and obtain the set of association rules, i.e., $\mathscr{R}$.
3: Initialize a fuzzy logic system with fuzzy rules $\mathscr{F}$.
4: **for** each word-pair (w_a, w_b) **do**
5: Computing local similarity S_{local} with $\mathbf{W}$ according to Eq.(6.8)
6: Computing global similarity S_{global} with the association rules in $\mathscr{R}$ according to Eq.(6.10)
7: Using the fuzzy membership function to quantify the membership degrees of two input variables, i.e., S_{local} and S_{global}, then using the fuzzy logic system to generate the fuzzy word similarity S_{fuzzy} between w_a and w_b.
8: **end for**
9: **return** the fuzzy word similarity between each word-pair in the vocabulary $\mathscr{V}$.

Table 63: The statistics for each dataset used in the experiments.

Datasets	Document Collection	Query ID	#Docs
TREC-6	TREC Disk 4&5	301-350	554,412
TREC-7	TREC Disk 4&5 without CR	351-400	528,155
TREC-8	TREC Disk 4&5 without CR	401-450	528,155
RCV1	Reuters Corpus Volume I	-	806,791
TREC-9	WT10G	451-500	1,692,096
TREC-10	WT10G	501-550	1,692,096

6.4 Experiments

We conducted experiments on the query expansion task to evaluate the proposed fuzzy word similarity measure and verify its usefulness in practice. The task setting is to compare different word similarity measures on a query expansion task. To better capture the users' real search intent, each query q is expanded with its top-k similar words to enrich the original query. Then, the information retrieval system leverages the expanded queries to search the candidate documents and return the retrieval results. A higher retrieval performance indicates a better similarity measure. Accordingly, our method is used to expand a user's search queries in real-world IR systems, with the local and global measures based on the collection of documents to be searched. The search engine leverages the extended set of queries to retrieve the relevant documents and return them to the user.

In this section, the used datasets and comparison baselines are described in Sections IV-A and IV-B, respectively. Section IV-C outlines the specific implementation steps taken in the experiments, followed by the results in Section IV-D. In Section IV-E, we discuss the method in terms of component validation, expansion size, and its parameters.

6.4.1 Datasets

Three widely used query expansion datasets are used in our experiments, including:

(1) **TREC-disk 4&5**. This dataset is designed to support research within the information retrieval community by providing the infrastructure necessary for large-scale

evaluation of text retrieval methodologies. It consists of news articles from various sources, including the Financial Times, the US Federal Register, the US Congressional Record, the Foreign Broadcast Information Service, and the LA Times. The dataset provides official queries, each of which is assigned a unique number. For instance, the 301 query is *International Organized Crime*, the 302 query is *Poliomyelitis and Post-Polio*. The dataset also provides relevance judgements for these queries against various portions of the TREC document collections, for evaluation. Within TREC-disk 4&5, we use three sets of queries[1], including:

- **TREC-6** is conducted on the whole document collection with query IDs 301-350.
- **TREC-7** is conducted on the TREC-disk 4&5 document collection except the Congressional Record with query IDs 351-400.
- **TREC-8** is conducted on the TREC-disk 4&5 collection without Congressional Record with query IDs 401-450.

(2) **Reuters Corpus Volume I (RCV1)**. This dataset is developed for the TREC filtering track. The first 50 collections are used in our experiments. Each collection has a topic statement file. We use topics, such as *Economic Espionage* and *Scottish Independence*, as the user-specified queries.

(3) **WT10G**. This is a relatively large document collection which consists of around 1.7 million web documents. Two official query sets are used in our experiments, including:

- **TREC-9** is conducted on the whole WT10G collection with query IDs 451-500.
- **TREC-10** is conducted on the WT10G collection with with query IDs 501-550.

Each document is described in the form of XML (Extensible Markup Language). We convert the XML documents into a series of plain text documents by removing any stop-words and converting all words to lower case. Then, we conduct term stemming using the Stanford Stemmer. The statistics of all datasets and queries are summarized in Table 63.

[1]The official queries are available in `https://trec.nist.gov/data/topics_eng/index.html`.

6.4.2 Baselines

The following baseline methods are used as comparisons:

- **WordNet**: A knowledge-based top-k selection that uses WordNet as an external resource to select similar words for a given query. Following Li et al. [196], we use WordNet to generate synonyms as the expansion words for each query.
- **AssoRule**: A corpus-based top-k selection method that uses association rules to select the expansion words. AssoRule is a variant of our proposed method that only selects queries using association rules. Only S_{global} is used to select the expansion words.
- **CBOW**[2] [8]: A word embedding-based top-k selection method. CBOW incorporates a neural network language model that learns word embeddings by maximizing the conditional probability of a target word given the context.
- **Skip-gram** [8]: A word embedding-based top-k selection method with a neural network language model that learns word embeddings by maximizing the conditional probability of a context word given the target word.
- **GloVe**[3] [7]: A word embedding-based top-k selection method that includes a state-of-the-art matrix factorization method that leverages global count information aggregated from the entire corpus as a word-word occurrence matrix to learn word embeddings.
- **LocalEmbed** [160]: A state-of-the-art word embedding-based query expansion method. This is a refinement of the typical word embedding methods that introduce local information into the query.
- **EnsemSim**: A word similarity ensemble method that makes use of word embeddings and knowledge bases (i.e., WordNet). In our experiments, two similarities are linearly combined using a combination parameter.

[2] `http://code.google.com/p/word2vec`
[3] http://nlp.stanford.edu/projects/glove/

- **FuzzySim** [190]: A fuzzy-rules-based word similarity method. FuzzySim re-weights the scores of the top-k words selected by word embeddings for the query expansion task.

6.4.3 Implementation and Baselines

In our experiments, given a query q, we first select several expansion words using different top-k selection methods. Then, the original query q together with its expansion words are used in an information retrieval model. Details of the experimental implementation and parameters are as follows.

Step 1: We generated the local and global measures from the corpus. The corpus is comprised of the top 1000 documents according to the initial retrieval results. Non-English words and stop words are discarded, then each word is converted to lower case. Word embeddings are trained using the Word2Vec[4] toolkit with the default settings. The dimensionality of word embeddings is 300 which is a typical setting in practice [197]. Then, the associated rules are mined using an association rule mining algorithm[5]. For each query, we select no more than 50 associated rules. It is notable that the selection of the threshold is easily dependent on the corpus, and in the used datasets, we manually set the confidence threshold of T_c and support threshold T_s to 0.8 and 10, respectively. The inputs, i.e., S_{local} and S_{global}, are generated in this step.

Step 2: Fuzzy word similarity measures are constructed using the fuzzy logic system. We applied Scikit-Fuzzy[6] to manage the fuzzy logic systems, which is a widely-used fuzzy logic toolkit. The range of the input and output variables is $[0, 1]$. Words with local scores below zero are irrelevant to the query and are not included. In this step, the fuzzy word similarity score S_{fuzzy} is learned from the inputs generated in *Step 1*.

Step 3: The query q is expanded by computing the similarity of q to each word in the vocabulary. The top-k similar words were selected as the expansion set, denoted as Q. Note that each of the top-k selection methods from the comparison baselines were tested

[4]http://code.google.com/p/word2vec
[5]https://github.com/bartdag/pymining
[6]https://pypi.python.org/pypi/scikit-fuzzy

in turn in this step, and each method returned different expansion sets for evaluation. The value of k is set to 5 in our experiments and its effectiveness with different settings is evaluated in Section 6.5.

Step 4: The expansion set in the information retrieval model is then evaluated. Within this model, the score of each document d for the query q and its expansion set Q is computed as:

$$Score(d,q,Q) = \lambda \sum_{q \in d} f(q) + (1-\lambda) \sum_{w \in Q, w' \in d} f(w) * S(w,q), \tag{6.11}$$

where $\lambda \in [0,1]$ is an interpolation parameter, $S(w,q)$ is the fuzzy word similarity between w and q, $f(w)$ is the weighting function of word w to document d. Specially, the BM25 scheme is applied. It is defined as:

$$f(w) = \frac{tf \cdot (a+1)}{a \cdot ((1-b) + b \cdot \frac{dl}{avgdl})} \cdot \log(\frac{N-n+0.5}{n+0.5}), \tag{6.12}$$

where N is the number of documents in the collection, n is the number of documents that contain term w, tf represents the term frequency, dl is the document length and $avgdl$ is the average document length. Following the suggested settings in Robertson et al. [198], a is set to 1.2 and b is set to 0.75.

For the implementation of the baseline methods: (1) the **WordNet** method selected no more than k synonyms for each query based on WordNet; (2) **AssoRule** is a fundamental component of the proposed method which only uses the global measure. We mined the association rules using the same settings as described in *Step 1*, and selected k words according to the global measure; (3) **CBOW**, **Skip-gram**, **GloVe** and **LocalEmbed** are embedding-based query expansion methods. We train these embeddings using the published code with default settings and selected the k most similar words for each query according to the words' cosine similarity; (4) **EnsemSim** uses word embeddings trained using CBOW and WordNet simultaneously, and the linear combination is set to 0.5; (5) **FuzzySim** defines fuzzy rules to reweight the embedding similarity. We use the four fuzzy rules defined in Liu et al. [190] and select k expansion words according to the reweighted similarities.

Table 64: Comparison of the proposed fuzzy word similarity measure with other similarity measures in terms of P@5, P@10, and MAP. Avg.Impr(%) is the average percentage of improvement of FWS over other baselines.

Dataset	Metric	LM	BM25	WordNet	AssoRule	CBOW	Skip-gram	GloVe	LocalEmbed	EnsemSim	FuzzySim	FWS
TREC-6	P@5	0.308	0.368	0.408	0.404	0.426	0.429	0.413	0.436	0.414	0.433	0.456
	P@10	0.288	0.318	0.364	0.372	0.376	0.373	0.369	0.382	0.374	0.377	0.394
	MAP	0.169	0.164	0.194	0.191	0.192	0.189	0.193	0.186	0.182	0.185	0.196
TREC-7	P@5	0.380	0.402	0.301	0.436	0.455	0.458	0.437	0.464	0.446	0.456	0.468
	P@10	0.354	0.380	0.268	0.398	0.417	0.415	0.394	0.426	0.408	0.417	0.430
	MAP	0.140	0.149	0.114	0.160	0.167	0.163	0.159	0.172	0.164	0.169	0.173
TREC-8	P@5	0.436	0.464	0.418	0.481	0.446	0.443	0.439	0.452	0.466	0.444	0.492
	P@10	0.420	0.424	0.376	0.442	0.445	0.449	0.447	0.440	0.446	0.443	0.458
	MAP	0.197	0.187	0.183	0.215	0.216	0.218	0.209	0.217	0.218	0.214	0.225
RCV1	P@5	0.436	0.578	0.532	0.568	0.571	0.573	0.568	0.581	0.569	0.573	0.613
	P@10	0.472	0.446	0.526	0.542	0.568	0.570	0.562	0.554	0.553	0.570	0.578
	MAP	0.419	0.408	0.436	0.439	0.448	0.449	0.449	0.442	0.441	0.448	0.451
TREC-9	P@5	0.271	0.282	0.288	0.292	0.291	0.290	0.278	0.296	0.294	0.291	0.315
	P@10	0.210	0.215	0.221	0.223	0.221	0.221	0.219	0.225	0.222	0.221	0.231
	MAP	0.141	0.145	0.149	0.155	0.153	0.154	0.147	0.156	0.154	0.151	0.163
TREC-10	P@5	0.332	0.344	0.336	0.340	0.351	0.348	0.345	0.344	0.343	0.350	0.361
	P@10	0.310	0.311	0.308	0.308	0.310	0.309	0.306	0.312	0.306	0.311	0.316
	MAP	0.145	0.151	0.143	0.148	0.153	0.148	0.145	0.150	0.148	0.151	0.163
Avg.Impr. (%)	P@5	$24.94^{\delta,\eta}$	$11.51^{\delta,\eta}$	19.50^{δ}	$7.41^{\delta,\eta}$	$6.44^{\delta,\eta}$	$6.48^{\delta,\eta}$	$9.24^{\delta,\eta}$	$5.19^{\delta,\eta}$	$6.82^{\delta,\eta}$	$6.17^{\delta,\eta}$	-
	P@10	16.95^{δ}	13.95^{δ}	17.92	$5.07^{\delta,\eta}$	$3.17^{\delta,\eta}$	$3.24^{\delta,\eta}$	$4.99^{\delta,\eta}$	2.74^{δ}	$4.24^{\delta,\eta}$	$3.10^{\delta,\eta}$	-
	MAP	$14.90^{\delta,\eta}$	$14.47^{\delta,\eta}$	17.09^{δ}	$5.57^{\delta,\eta}$	$3.93^{\delta,\eta}$	$4.91^{\delta,\eta}$	6.96^{δ}	$4.14^{\delta,\eta}$	$5.87^{\delta,\eta}$	$5.02^{\delta,\eta}$	-

The official evaluation metrics, i.e., the mean average precision (MAP) for the top 1000 documents, the precision at 5 (P@5), and the precision at 10 (P@10) are reported for comparison purposes.

6.4.4 Results

The overall performance is presented in Table 64. The language model (LM) [199] and BM25 are basic information retrieval methods that do not include query expansion. In the language model method, the probability of producing queries given a document d using a maximum likelihood estimation under the unigram assumption is calculated as the relevance ranking of d: $rank(d,S) = \prod_{s \in S} \frac{tf_{s,d}}{L_d}$, where $tf_{s,d}$ is the term frequency of the query s in document d, and L_d is the number of tokens in document d The BM25 method evaluates the similarity between the query and the document candidates using Eq.(6.12). The last group is the percentage of the average improvement of the FWS method over other baseline methods. To decide whether the improvement of our method over the baseline methods is significant, we conduct a t-test to calculate a value p based

on the performance of FWS and the baseline method. The smaller the value of p, the more significant the improvement. If the value of p is small enough, we conclude that the improvement is statistically significant. The superscript δ and η denote statistically significant improvements over baseline methods with $p < 0.05$ and $p < 0.01$, respectively. Table 64 shows our method significantly outperforms the baseline methods, including word embedding methods and association rule methods.

Furthermore, most methods systematically obtain better results than LM and BM25, which indicates that the top-k selection methods are more effective in discovering words similar to the original query and improved information retrieval performance. This result is unsurprising since these basic methods suffer from a well-known vocabulary mismatch problem.

We also observe that the WordNet method does not perform as well as the classical methods for the TREC-7 and TREC-8 datasets. The main reason for this is that the WordNet method selects expansion words based on human-defined relations without task-specific information. The AssoRule method demonstrates better performance than the classical methods, suggesting that association rules are useful for query expansion. However, the AssoRule method does not yield any obvious advantages over the methods based on word embedding. A possible reason for this is that the size of the association rules is large, and as such, more noise could be introduced.

The proposed fuzzy word similarity measure consistently outperforms all baseline methods across all datasets. In particular, our method shows an improvement in MAP and P@5 over the best results of the baselines. These results indicate that the proposed method is both highly effective and reliable on query expansion tasks. Our method also outperforms CBOW, Skip-gram, and GloVe on all datasets, especially TREC-6 and TREC-7. This observation shows the superiority of our method and demonstrates the effectiveness of jointly using both local and global measures for top-k selection tasks. The LocalEmbed method is the most competitive baseline. According to Table 64, our method performs better than LocalEmbed on all datasets, mainly because word similarity in our method is learned from the local measure plus related words over a long

distance according to the association rules. Our method also outperforms EnsemSim, which is a simple ensemble word similarity method, indicating the advantage of the fuzzy combination method. FuzzySim designed four fuzzy rules to reweight the word embedding similarities, while our method designed a fuzzy logic system to use the word embedding and association rules jointly. Compared with the FuzzySim method, our method achieves better results, indicating that the complementary global information is essential to improve the query expansion performance.

Notably, our method outperforms the other baselines by a large margin on the RCV1 dataset. As shown in Table 63, RCV1 is a small dataset compared with TREC. Generally, small datasets suffer from significant vocabulary mismatching problems, and query expansion methods play a vital role in improving retrieval performance in these situations. Moreover, traditional methods that select expansion terms from the corpus often suffer data sparsity problems. Our method delivers better performance in this scenario, which illustrates the effectiveness of our method in selecting related terms when applied to a small dataset.

6.5 In-depth Analysis

6.5.1 Component-wise Validation

Our method contains three components, i.e., local measure, global measure, and a fuzzy logic system. To gain a better understanding of each component, we conduct a component-wise validation by comparing the performance of different combinations of each component. We test four variants as follows:

- Case 1: only the local component, with the expansion words selected from the vocabulary using S_{local}.

- Case 2: only the global component, with the expansion words selected from the association rules set $\mathscr{R}$ using S_{global}.

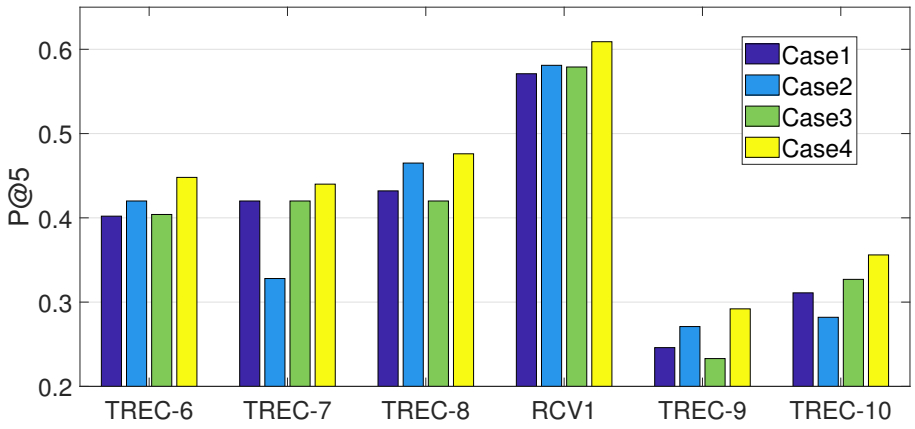
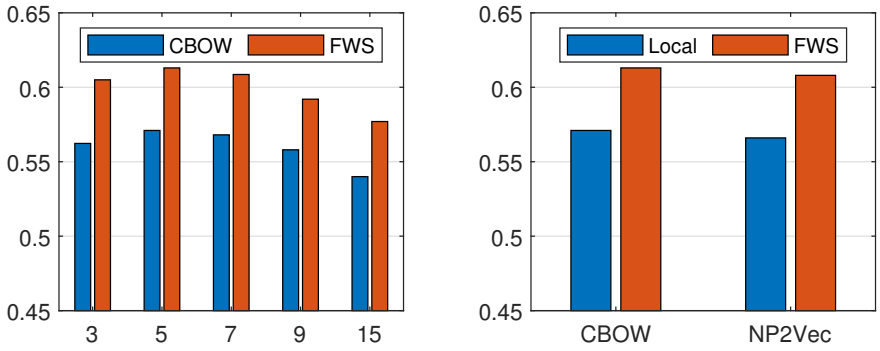


Figure 63: The component-wise validation performance for each dataset

- Case 3: both the local and global components, with the top $k/2$ words selected separately, then directly combined. The fuzzy logic system is not included in this case.

- Case 4: our complete method with all three components (i.e., local measure, global measure, and fuzzy logic system). Unlike Case 3, all candidates are re-ranked using the fuzzy word similarity measures.

Figure 63 reports the performance results of all four cases. In Case 1, we observe that ablating the global measure leads to a dramatic performance drop compared with the full method in Case 4, indicating that long-term co-occurrence information discovered by the global measure is effective for query expansion. The low performance of Case 2 denotes that the local measure computed by word embeddings is also essential to question. The above phenomena confirm the effectiveness of jointly leveraging both local and global measures. In addition, it is observed that Case 3 does not achieve significant improvements compared with Case 1 and Case 2, showing that directly combining the local and global components may introduce more noisy data and damage the effect of query expansion in information retrieval. The full method Case 4 performed better than Case 3, verifying the effectiveness of the fuzzy scheme and its ability to balance the individual contributions of the local and global components.



(a) context window size (b) local measure

Figure 64: The influence of the local measure with different settings in terms of P@5 on the RCV1 dataset. (a) A comparison with varying context window sizes. (b) A comparison between the original Word2Vec model and the its variation NP2Vec, and *Local* denotes the query expansion method with only a local measure.

6.5.2 Local Measure

In this subsection, we verify the impact of using different settings for the local measure on the FWS method. The experiments are conducted on the RCV1 dataset.

First, we evaluate the impact of different context window sizes. We use different window sizes in the range $\{3,5,9,15\}$ to train different word embeddings. These different embeddings are employed as a local measure in the proposed FWS method. Their performance on the RCV1 dataset is compared in Fig. 64 (a). It is observed that the CBOW method achieves the best performance when the window size is 5. When the window size is set to 15, the performance decreases significantly, indicating that increasing the window size may introduce more noise words and weaken the ability of the learned embedding space to detect related words. We observe that the proposed FWS method achieves stable performance gains over the CBOW method, and high-quality local measures improve the quality of our method.

Then, we evaluate the impact of using noun phrases (NP) to vector (denoted as NP2Vec[7]) as a local measure in the FWS method. NP2Vec is a variation of the Word2Vec method, which assumes that the NPs are already marked in the input corpus and learns

[7] `https://github.com/NervanaSystems/nlp-architect/tree/master/examples/np2vec`

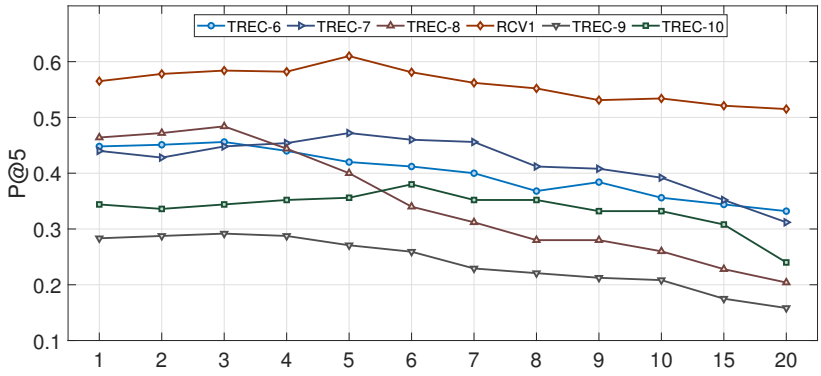


Figure 65: The effect of varying the size of the query expansion in terms of P@5 with different datasets.

the vectors of NPs. In implementation, we leverage the NLTK toolkit to detect NPs. The character "_" is used as the connection, i.e., the NP *query expansion* is marked as *query_expansion*. We use cosine similarity to select the top similar terms to the query. If the selected terms include NPs, we split them into words. The performance of using CBOW and NP2Vec is shown in Fig. 64 (b). We observe that when using NP2Vec as the local measure, the FWS method achieves significant improvement, indicating its stability and superiority when using different local measures. NP2Vec leverages the phrases which are more comprehensive and informative than words. However, we observe that NP2Vec achieved a similar result with CBOW. There are two possible reasons: (1) both NP2Vec and CBOW use context information to learn the semantic features, resulting in a similar performance; (2) the phrase of query[8] may be an out-of-vocabulary term in the corpus.

6.5.3 Expansion Size

In our method, the top-k candidate words are selected according to their fuzzy word similarity score. Hence, we evaluate the performance of all baselines with different expansion sizes k. The results are shown in Figure 65.

[8]For the query which contains multiple words, we combine them as a phrase and search its most similar terms using NP2Vec. When it is not in the vocabulary, we split it into words and separately collect the most similar terms.

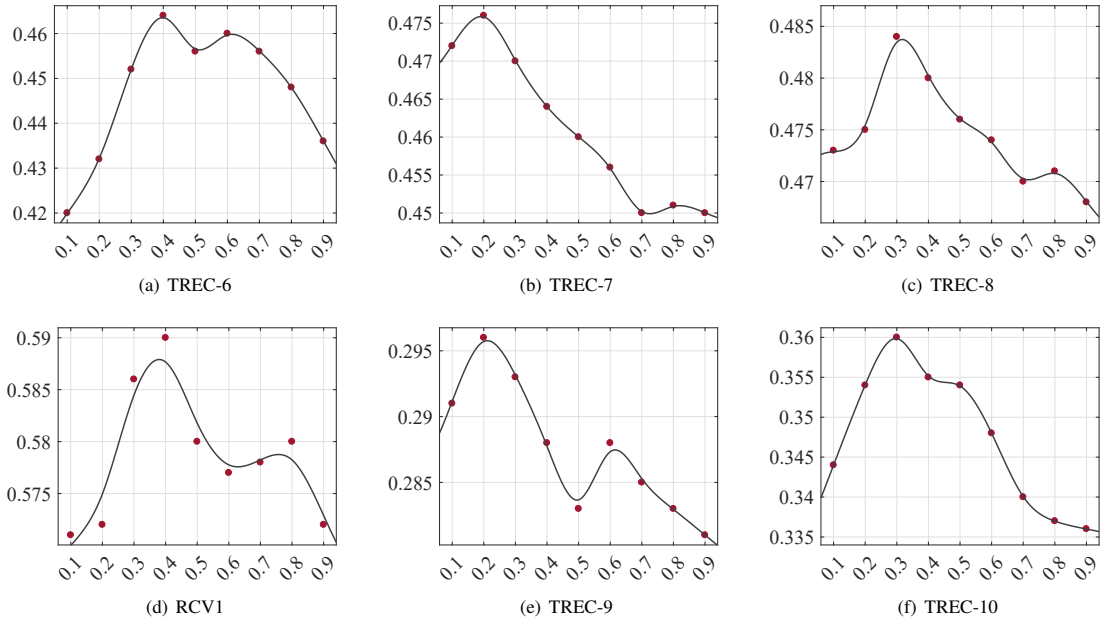


Figure 66: The performance with a varying λ on five development datasets. The Y-axis represents P@5, and the X-axis represents different λ.

Our method delivers the best performance when the queries are expanded by three words in TREC-6, TREC-8, TREC-9. 5 and 6 words are needed for the peak performance in TREC-7 and TREC-10. A large number of expansion terms significant decrease the performance for all datasets. This decreased performance is because the original queries are short (some only comprising one or two terms) and therefore the addition of many more terms may have introduced more noise.

6.5.4 Parameter λ

Parameter λ in Eq.(6.11) regulates the effects of the original queries and expansion queries. To evaluate the sensitivity of this parameter, we tune λ using a grid search to between 0.1 to 1.0 in steps of 0.1 for our method with the different datasets. The results are shown in Figure 66. Here, we observe that a smaller λ results in a better performance with all datasets. For example, on the TREC-7 and TREC-9 datasets, the optimal $P@5$ scores are achieved when λ equals 0.2. On TREC-6, TREC-8, and TREC-10, the best performance is obtained with values of 0.4, 0.3, and 0.3, respectively. This observation

Table 65: The Top-ten selected words for five queries using the original word embedding method (denoted as *Original*) and the fuzzy word similarity measure (denoted as *FWS*) on the RCV1 dataset. The words in bold are contributed by the global measure, and the followed score is S_{global}.

insurance		software		license		map		power	
Original	FWS	Original	FWS	Original	FWS	Original	FWS	Original	FWS
corporate	corporate	hypercard	linux	LGPL	LGPL	geography	geography	reactor	reactor
private	**health**(0.96)	linux	**developed**(0.96)	GPL	**renewal**(0.94)	coordinates	topographic	control	electric
liability	**policy**(0.92)	developers	**technology**(0.88)	licence	freeware	geographic	**miles**(0.98)	majorities	**lasers**(0.92)
annuities	private	unix	hardware	GNU	GPL	topographic	coordinates	electric	energy
pensions	**longterm**(0.86)	mozilla	**design**(0.82)	FSF	**byproduct**(0.9)	location	geographic	advantage	**control**(0.84)
banking	**care**(0.78)	proprietary	developers	BSD	GNU	atlas	**acreage**(0.92)	energy	majorities
business	liability	windows	windows	licensing	licence	clickable	geographic	mox	fuel
charitable	annuities	hardware	**producing**(0.84)	freeware	software	rectangle	location	leadership	advantage
securities	**coverage**(0.9)	direct	proprietary	software	copyleft	arcology	**desert**(0.94)	fuel	continuous
management	pensions	gratis	unix	copyleft	**inspections**(0.92)	mapping	**road**(0.88)	flywheel	**plus**(0.86)

shows that a smaller weighting for the expansion set can be used to generate a more powerful information retrieval model. However, performance declines as the value of λ increases, as this leads to too much information loss from the original query. Therefore, our recommended setting for λ in the proposed method falls with a range of 0.2 to 0.4.

6.5.5 Case Study

In this section, we describe a case study on the RCV1 dataset to provide in-depth insights into the proposed fuzzy word similarity measure. To be specific, we compare the top-ten words selected for *insurance*, *software*, *license*, *map*, and *power* using a traditional word embedding method and the proposed FWS method.

The selected top-ten words are shown in Table 65. The words in bold are selected by the global measure, and their global score S_{global} is shown in parentheses. We observe that the proposed fuzzy word similarity measure returns a diverse set of related words. For example, *developed*, *design*, and *producing* are related to *software*; *miles*, *acreage*, and *road* are related to *map*. These words have high global scores, showing they are easily detected by association rules. On the other hand, they are ignored by the word embedding method. A possible reason is that they co-occur with the query word with a long distance in the corpus. These observations reveal a unique advantage of our method, in that the global measure can be considered by mining association rules.

An interesting future study to further improve the performance of the proposed

method is to control the semantic consistency between the query words and expanded words . According to our observation, most error cases of the FWS method occurred as a result of the ignorance of the relationships between words in a query because few rules contain the patterns which exactly match the query. For example, the query *software engineer* is incorrectly extended by the word *producing* which is related to *software* but not to *engineer*. To alleviate this drawback, it is necessary to understand the query as a whole term, not as a combination of words.

6.6 Summary

This chapter presents a fuzzy word similarity measure for top-k words selection that jointly assesses both local measure and global measure in a corpus. A word embedding method measures the similarity between the words extracted from local information, and association rules are used to measure the relatedness between the words mined from global information. A fuzzy logic system overcomes the problems associated with combining the two types of measures by inferring the similarity between words, then returns the top-k selected words. We evaluated the proposed method on a query expansion task with six datasets from three widely used document collections: TREC-disk 4&5, WT10G, and RCV1. The results demonstrate the excellent strength of our method compared to several state-of-the-art baselines.

CHAPTER 7

CONCLUSION AND FUTURE STUDY

This chapter concludes the thesis and provides several further research directions for word representation with transferable semantics.

7.1 Conclusions

The development of artificial intelligence generates an increasing demand for a machine to understand natural languages. Hence, as the first step in converting natural languages to a machine-processable format, semantic representation learning is attracting increasing attention. Most research mainly leverage raw corpora to encode semantic knowledge without considering the knowledge existing in other resources. In this thesis, we consider how to leverage semantic knowledge from various domains.

To sum up, existing semantic representation learning methods still face the following problems in the real world: 1) how to reliably transfer semantics from a structural knowledge base to an unstructured representation space; 2) how to reliably transfer semantics from multiple source domains to a low-resource target domain; 3) how to achieve the reliable and low-cost cross-lingual transfer of semantics; and 4) how to adapt semantic representations for specific applications.

To solve the aforementioned challenges, this thesis proposed four research questions and corresponding research objectives. The findings of this study are summarized as follows:

1. *We design the assumptions of concept divergence and word convergence to model semantic structures in knowledge bases and propose the semantic structure-based method to learn knowledge-enhanced word representations.* (to achieve RO1)

   Text and knowledge bases are complementary sources for word representation. Most existing methods only consider the relationships within word-pairs in the use of knowledge bases. We argue that the structural information of well-organized words within the knowledge base conveys more effective and stable knowledge in capturing the semantics of words. In this thesis, we propose a semantic structure-based word embedding method, and introduce concept convergence and word divergence to reveal semantic structures in the word embedding learning process. To assess the effectiveness of our method, we use WordNet for training and conduct extensive experiments on word similarity, word analogy, text classification and query expansion. The experiment results show that our method outperforms the state-of-the-art methods, including the methods trained solely on the corpus, and others trained on the corpus and the knowledge base.

2. *We propose a new meta-embedding method to dynamically leverage semantics from multiple source domains.* (to achieve RO2)

   Meta-embedding aims at assembling pre-trained embeddings from various sources and producing more expressively powerful word representations. Many natural language processing tasks in a specific domain benefit from meta-embedding, especially when the task suffers from low resources. This thesis proposes an unsupervised meta-embedding method that jointly models background knowledge from the source embeddings and domain-specific knowledge from the task domain. Specifically, embeddings from multiple sources for a word are dynamically aggregated to a single meta-embedding by a differentiable attention module. The

embeddings derived from pre-training on a large-scale corpus provide the complete background knowledge of word usage. Then, the meta-embedding is further enriched by exploring domain-specific knowledge from each task domain in two ways. First, contextual information in the raw corpus is considered to capture the semantics of words. Second, a graph representing domain-specific semantic structures is extracted from the raw corpus to highlight the relationships between salient words, then the graph is modeled by a powerful graph convolution network to effectively capture the rich semantic structures among words in the task domain. Experiments conducted on two tasks, i.e., text classification and relation extraction, show that our model outputs more accurate word meta-embeddings for the task domain, compared to other state-of-the-art competitors.

3. *We propose a pivot-based method to bridge the cross-lingual semantic gap with limited bilingual resource reliance.* (to achieve RO3)

   We propose a pivot-based approach to bridge the cross-lingual semantic gap. It takes an intermediary set of plausible target-language mentions as pivots for the cross-lingual entity candidate retrieval task. It first converts mentions in the source language into an intermediary set of plausible mentions in the target language by cross-lingual semantic retrieval and a selective mechanism, and then retrieves candidate entities based on the generated mentions by lexical retrieval. Thus, our method employs multilingual embeddings to apply English retrieval model on other languages. The proposed approach only relies on a small bilingual word dictionary and fully exploits the benefits of both lexical and semantic matching. The experiment results on two challenging cross-lingual entity linking datasets spanning over 11 languages show that the pivot-based approach outperforms both the lexicon-based and semantic-based approach by a large margin.

4. *We propose a fuzzy semantic measure to adapt general semantic representations according to the task-oriented features.* (to achieve RO4)

   We consider the top-k words selection task, which is a technique used to detect

and return the k most similar words to a given word from a candidate set. The key issue in top-k words selection is how to measure the similarity between words. One popular and effective solution is to use a word embedding-based similarity measure, which represents words as low-dimensional vectors and measures the similarities between words according to the similarity of the vectors, using a metric. However, most word representation methods only consider the local proximity properties of two words in a corpus. To mitigate this issue, we propose to refine embedding similarity by using association rules for measuring word similarity at a global level, and a fuzzy similarity measure for top-k words selection that jointly encodes the local and global similarities. Experiments on a real-world query task with three benchmark datasets demonstrate the efficiency of the proposed method compared to several state-of-the-art baselines.

7.2 Future Study

This thesis identifies the following directions as future work:

1. *Improving pre-trained language models with transferable semantics from the knowledge base.*

   Pre-trained language models are contextual representation methods which can dynamically capture the semantics of words according to their context. They are pre-trained on a large-scale corpus and fine-tuned in the downstream tasks. Knowledge bases and corpora are complementary to capture the semantics of textual data. In this thesis, we explore how to encode semantic structures from knowledge bases into word representations. In the future, we aim to propose a method to enhance the pre-trained language model with semantic structures in the knowledge bases.

2. *A new salient word selection for integrating multiple source embeddings.*

   We proposed a dynamical meta-embedding to combine multiple source embeddings to learn semantic representation for a specific domain. We employed a statistic-

based method to select salient words for the specific domain, to capture the domain-specific features. To further improve the performance, in the future, we will design an automatic salient word selection method for a specific domain.

3. *A new method to bridge the cross-lingual semantic gap for sentence-level tasks.*

   We proposed a pivot-based method to bridge the cross-lingual semantic gap for cross-lingual retrieval. These pivots are plausible English mentions which semantically related to the queries in other languages. This ensures that queries in other languages can be processed using the English retrieval model. Most queries consist of a few words. We treat each word equally and use linear combination to generate pivots. In the future, we will improve the quality of the pivot set by automatically detecting the key-phrase of the source-language mention and alleviating the out-of-vocabulary problem.

4. *More industry-level applications of the proposed methods.*

   In this thesis, the proposed methods are successfully used to solve many real-world problems (such as text classification, query expansion and cross-lingual entity linking). In the future, we still need to develop more prototypes of these methods and apply these prototypes in industry. Moreover, the proposed methods will be used to address more real-world problems in the field of knowledge systems and business dialogue systems.

BIBLIOGRAPHY

[1] Z. Liu, Y. Lin, and M. Sun, *Representation Learning for Natural Language Processing*. Springer, 2020.

[2] N. Rekabsaz, M. Lupu, A. Hanbury, and H. Zamani, "Word embedding causes topic shifting; exploit global context!" in *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2017, pp. 1105–1108.

[3] X. Li, M. de Rijke, Y. Liu, J. Mao, W. Ma, M. Zhang, and S. Ma, "Learning better representations for neural information retrieval with graph information," in *Conference on Information and Knowledge Management (CIKM)*, 2020, pp. 795–804.

[4] L. M. Antony and M. Davies, "Meaning and semantic knowledge," *Proceedings of the Aristotelian Society, Supplementary Volumes*, vol. 71, pp. 177–209, 1997.

[5] J. R. Firth, "A synopsis of linguistic theory 1930-55," *Selected Papers of J. R. Firth (1952-59)*, vol. 1952-59, pp. 168–205, 1957.

[6] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[7] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[8] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2013, pp. 3111–3119.

[9] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018, pp. 2227–2237.

[10] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019, pp. 4171–4186.

[11] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019.

[12] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *CoRR*, vol. abs/1906.08237, 2019.

[13] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[14] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: An open multilingual graph of general knowledge," in *AAAI Conference on Artificial Intelligence (AAAI)*, S. P. Singh and S. Markovitch, Eds., 2017, pp. 4444–4451.

[15] Y. Liu, Z. Liu, T. Chua, and M. Sun, "Topical word embeddings," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2015, pp. 2418–2424.

[16] Q. Liu, H. Huang, Y. Gao, X. Wei, and R. Geng, "Leveraging pattern associations for word embedding models," in *Database Systems for Advanced Applications*, vol. 10177, 2017, pp. 423–438.

[17] Q. Liu, H. Huang, Y. Gao, X. Wei, Y. Tian, and L. Liu, "Task-oriented word embedding for text classification," in *International Conference on Computational Linguistics (COLING)*, 2018, pp. 2023–2032.

[18] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for twitter sentiment classification," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2014, pp. 1555–1565.

[19] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence: A survey," *Knowledge-Based Systems*, vol. 80, pp. 14–23, 2015.

[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013.

[21] Y. Tsuboi, "Neural networks leverage corpus-wide information for part-of-speech tagging," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 938–950.

[22] W. Yin and H. Schütze, "Learning word meta-embeddings," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016, pp. 1351–1360.

[23] J. Coates and D. Bollegala, "Frustratingly easy meta-embedding - computing meta-embeddings by averaging source word embeddings," in *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018, pp. 194–198.

[24] C. Bao and D. Bollegala, "Learning word meta-embeddings by autoencoding," in *International Conference on Computational Linguistics (COLING)*, 2018, pp. 1650–1661.

[25] C. Tsai and D. Roth, "Cross-lingual wikification using multilingual embeddings," in *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2016, pp. 589–598.

[26] S. Rijhwani, J. Xie, G. Neubig, and J. G. Carbonell, "Zero-shot neural transfer for cross-lingual entity linking," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2019, pp. 6924–6931.

[27] Y. Ren, Y. Zhang, M. Zhang, and D. Ji, "Improving twitter sentiment classification using topic-enriched multi-prototype word embeddings," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2016, pp. 3038–3044.

[28] H. Zamani and W. B. Croft, "Relevance-based word embedding," in *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2017, pp. 505–514.

[29] M. Yu and M. Dredze, "Improving lexical embeddings with semantic knowledge," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2014, pp. 545–550.

[30] D. Bollegala, M. Alsuhaibani, T. Maehara, and K. Kawarabayashi, "Joint word representation learning using a corpus and a semantic lexicon," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2016, pp. 2690–2696.

[31] Y. Chen, B. Perozzi, R. Al-Rfou', and S. Skiena, "The expressive power of word embeddings," *International Conference on Machine Learning (ICML), Workshop*, 2013.

[32] S. Lai, K. Liu, S. He, and J. Zhao, "How to generate a good word embedding," *IEEE Intelligent Systems*, vol. 31, no. 6, pp. 5–14, 2016.

[33] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, vol. 60, no. 5, pp. 493–502, 2004.

[34] X. Pan, B. Zhang, J. May, J. Nothman, K. Knight, and H. Ji, "Cross-lingual name tagging and linking for 282 languages," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017, pp. 1946–1958.

[35] G. Lample, A. Conneau, M. Ranzato, L. Denoyer, and H. Jégou, "Word translation without parallel data," in *International Conference on Learning Representations (ICLR)*, 2018.

[36] Z. S. Harris, "Distributional structure," *Word-journal of The International Linguistic Association*, vol. 10, no. 2-3, pp. 146–162, 1954.

[37] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research (JMLR)*, vol. 3, pp. 993–1022, 2003.

[38] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *Journal of Machine Learning Research (JMLR)*, vol. 3, pp. 1137–1155, 2003.

[39] A. Mnih and G. E. Hinton, "A scalable hierarchical distributed language model," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2008, pp. 1081–1088.

[40] M. Artetxe, G. Labaka, and E. Agirre, "Learning bilingual word embeddings with (almost) no bilingual data," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017, pp. 451–462.

[41] O. Barkan, "Bayesian neural word embedding," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2017, pp. 3135–3143.

[42] C. Li, L. Ji, and J. Yan, "Acronym disambiguation using word embedding," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2015, pp. 4178–4179.

[43] B. Shi, W. Lam, S. Jameel, S. Schockaert, and K. P. Lai, "Jointly learning word embeddings and latent topics," in *Annual International ACM SIGIR Conference*

on Research and Development in Information Retrieval (SIGIR), 2017, pp. 375–384.

[44] P. K. Sarma, Y. Liang, and B. Sethares, "Domain adapted word embeddings for improved sentiment classification," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018, pp. 37–42.

[45] G. Wang, C. Li, W. Wang, Y. Zhang, D. Shen, X. Zhang, R. Henao, and L. Carin, "Joint embedding of words and labels for text classification," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018, pp. 2321–2331.

[46] A. Roy and S. Pan, "Incorporating extra knowledge to enhance word embedding," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2020, pp. 4929–4935.

[47] D. Bollegala, T. Maehara, and K. Kawarabayashi, "Embedding semantic relations into word representations," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2015, pp. 1222–1228.

[48] M. Faruqui, J. Dodge, S. K. Jauhar, C. Dyer, E. Hovy, and N. A. Smith, "Retrofitting word vectors to semantic lexicons," in *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2015, pp. 1606–1615.

[49] J. Ganitkevitch, B. V. Durme, and C. Callison-Burch, "PPDB: the paraphrase database," in *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics*, 2013, pp. 758–764.

[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008.

[51] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[52] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[53] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2020, pp. 1877–1901.

[54] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," in *International Conference on Learning Representations (ICLR)*, 2020.

[55] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "ERNIE: enhanced language representation with informative entities," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019, pp. 1441–1451.

[56] M. E. Peters, M. Neumann, R. L. L. IV, R. Schwartz, V. Joshi, S. Singh, and N. A. Smith, "Knowledge enhanced contextual word representations," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019, pp. 43–54.

[57] J. Zhou, Z. Zhang, H. Zhao, and S. Zhang, "LIMIT-BERT : Linguistics informed multi-task BERT," in *Conference on Empirical Methods in Natural Language Processing (EMNLP): Findings*, 2020, pp. 4450–4461.

[58] Y. Levine, B. Lenz, O. Dagan, O. Ram, D. Padnos, O. Sharir, S. Shalev-Shwartz, A. Shashua, and Y. Shoham, "Sensebert: Driving some sense into BERT," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020, pp. 4656–4667.

[59] W. Liu, P. Zhou, Z. Zhao, Z. Wang, Q. Ju, H. Deng, and P. Wang, "K-BERT: enabling language representation with knowledge graph," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2020, pp. 2901–2908.

[60] P. Ke, H. Ji, S. Liu, X. Zhu, and M. Huang, "Sentilr: Linguistic knowledge enhanced language representation for sentiment analysis," *CoRR*, vol. abs/1911.02493, 2019.

[61] X. Wang, T. Gao, Z. Zhu, Z. Liu, J. Li, and J. Tang, "KEPLER: A unified model for knowledge embedding and pre-trained language representation," *CoRR*, vol. abs/1911.06136, 2019.

[62] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-trained models for natural language processing: A survey," *CoRR*, vol. abs/2003.08271, 2020.

[63] M. Chen, K. Q. Weinberger, and J. Blitzer, "Co-training for domain adaptation," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2011, pp. 2456–2464.

[64] W. Dai, Y. Chen, G. Xue, Q. Yang, and Y. Yu, "Translated learning: Transfer learning across different feature spaces," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2008, pp. 353–360.

[65] P. Prettenhofer and B. Stein, "Cross-language text classification using structural correspondence learning," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2010, pp. 1118–1127.

[66] L. Wang, L. Liu, and L. Zhou, "A graph-embedding approach to hierarchical visual word mergence," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 2, pp. 308–320, 2017.

[67] X. Zheng, J. Feng, Y. Chen, H. Peng, and W. Zhang, "Learning context-specific word/character embeddings," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2017, pp. 3393–3399.

[68] J. Goikoetxea, E. Agirre, and A. Soroa, "Single or multiple? combining word representations independently learned from text and wordnet," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2016, pp. 2608–2614.

[69] C. Xu, Y. Bai, J. Bian, B. Gao, G. Wang, X. Liu, and T. Liu, "RC-NET: A general framework for incorporating knowledge into word representations," in *Conference on Information and Knowledge Management (CIKM)*, 2014, p. 1219,Äì1228.

[70] J. Grover and P. Mitra, "Bilingual word embeddings with bucketed CNN for parallel sentence extraction," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017, pp. 11–16.

[71] F. Aydin, Z. M. Hüsünbeyi, and A. Özgür, "Automatic query generation using word embeddings for retrieving passages describing experimental methods," *Database*, vol. 2017, 01 2017, baw166.

[72] L. Yao, Y. Zhang, Q. Chen, H. Qian, B. Wei, and Z. Hu, "Mining coherent topics in documents using word embeddings and large-scale text data," *Application of AI*, vol. 64, pp. 432–439, 2017.

[73] C. Tao, M. Filannino, and Ö. Uzuner, "Prescription extraction using crfs and word embeddings," *Journal of Biomedical Informatics*, vol. 72, pp. 60–66, 2017.

[74] X. Yang and K. Mao, "Task independent fine tuning for word embeddings," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 25, no. 4, pp. 885–894, 2017.

[75] S. Kuzi, A. Shtok, and O. Kurland, "Query expansion using word embeddings," in *Conference on Information and Knowledge Management (CIKM)*, 2016, pp. 1929–1932.

[76] D. Ganguly, D. Roy, M. Mitra, and G. J. F. Jones, "Word embedding based generalized language model for information retrieval," in *Annual International*

ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 2015, pp. 795–798.

[77] S. Balaneshinkordan and A. Kotov, "Embedding-based query expansion for weighted sequential dependence retrieval model," in *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2017, pp. 1213–1216.

[78] P. Arora, J. Foster, and G. J. F. Jones, "Query expansion for sentence retrieval using pseudo relevance feedback and word embedding," in *Conference and Labs of the Evaluation Forum / Workshop of Cross-Language Evaluation Forum*, 2017, pp. 97–103.

[79] J. Karlgren, A. Holst, and M. Sahlgren, "Filaments of meaning in word space," in *European Conference on Information Retrieval (ECML)*, 2008, pp. 531–538.

[80] N. Rekabsaz, M. Lupu, and A. Hanbury, "Exploration of a threshold for similarity based on uncertainty in word embedding," in *European Conference on Information Retrieval (ECML)*, 2017, pp. 396–409.

[81] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 35, no. 8, pp. 1798–1828, 2013.

[82] N. Passalis and A. Tefas, "Entropy optimized feature-based bag-of-words representation for information retrieval," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1664–1677, 2016.

[83] D. Roy, D. Ganguly, M. Mitra, and G. J. Jones, "Word vector compositionality based relevance feedback using kernel density estimation," in *Conference on Information and Knowledge Management (CIKM)*, 2016.

[84] V. Lampos, B. Zou, and I. J. Cox, "Enhancing feature selection using word embeddings: The case of flu surveillance," in *The Web Conference (WWW)*, 2017, pp. 695–704.

[85] Y. Shen, W. Rong, N. Jiang, B. Peng, J. Tang, and Z. Xiong, "Word embedding based correlation model for question/answer matching," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2017, pp. 3511–3517.

[86] D. Bollegala, T. Mu, and J. Y. Goulermas, "Cross-domain sentiment classification using sentiment sensitive embeddings," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 398–410, 2016.

[87] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research (JMLR)*, vol. 12, pp. 2493–2537, 2011.

[88] S. Cao and W. Lu, "Improving word embeddings with convolutional feature learning and subword information," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2017, pp. 3144–3151.

[89] S. P. Ponzetto and R. Navigli, "Knowledge-rich word sense disambiguation rivaling supervised systems," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2010, pp. 1522–1531.

[90] Q. Liu, H. Jiang, S. Wei, Z. Ling, and Y. Hu, "Learning semantic word embeddings based on ordinal knowledge constraints," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2015, pp. 1501–1511.

[91] J. Xuan, X. Luo, G. Zhang, J. Lu, and Z. Xu, "Uncertainty analysis for the keyword system of web events," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 6, pp. 829–842, 2016.

[92] J. Xuan, J. Lu, G. Zhang, R. Y. D. Xu, and X. Luo, "Bayesian nonparametric relational topic model through dependent gamma processes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 7, pp. 1357–1369, 2017.

[93] R. Lebret and R. Collobert, "Word embeddings through hellinger PCA," in *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2014, pp. 482–490.

[94] D. Bollegala, T. Maehara, Y. Yoshida, and K. Kawarabayashi, "Learning word representations from relational graphs," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

[95] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *ACM SIGMOD Conference (SIGMOD)*, 2008, pp. 1247–1250.

[96] J. Ganitkevitch, B. V. Durme, and C. Callison-Burch, "PPDB: The paraphrase database," in *North American Chapter of the Association for Computational Linguistics (NAACL)*, June 2013, pp. 758–764.

[97] G. A. Miller and W. G. Charles, "Contextual correlates of semantic similarity," *Language and Cognitive Processes*, vol. 6, pp. 1–28, 1991.

[98] E. Bruni, G. Boleda, M. Baroni, and N. Tran, "Distributional semantics in technicolor," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2012.

[99] T. Luong, R. Socher, and C. D. Manning, "Better word representations with recursive neural networks for morphology," in *Conference on Computational Natural Language Learning (CoNLL)*, 2013.

[100] S. Baker, R. Reichart, and A. Korhonen, "An unsupervised model for instance level subcategorization acquisition," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[101] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Paşca, and A. Soroa, "A study on similarity and relatedness using distributional and wordnet-based approaches," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2009.

[102] C. Spearman, "The proof and measurement of association between two things," *The American Journal of Psychology*, vol. 15, pp. 72–101, 1904.

[103] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International Conference on Machine Learning (ICML)*, 2014, pp. 1188–1196.

[104] D. Bollegala, K. Hayashi, and K. Kawarabayashi, "Think globally, embed locally - locally linear meta-embedding of words," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018, pp. 3970–3976.

[105] W. Yang, W. Lu, and V. Zheng, "A simple regularization-based algorithm for learning cross-domain word embeddings," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017, pp. 2898–2904.

[106] J. Lu, H. Zuo, and G. Zhang, "Fuzzy multiple-source transfer learning," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 12, pp. 3418–3431, 2020.

[107] D. Kiela, C. Wang, and K. Cho, "Dynamic meta-embeddings for improved sentence representations," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018, pp. 1466–1477.

[108] H. Xu, B. Liu, L. Shu, and P. S. Yu, "Lifelong domain word embedding via meta-learning," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018, pp. 4510–4516.

[109] A. Hazem and E. Morin, "Leveraging meta-embeddings for bilingual lexicon extraction from specialized comparable corpora," in *International Conference on Computational Linguistics (COLING)*, 2018, pp. 937–949.

[110] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2015, pp. 2224–2232.

[111] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[112] W. Wang, X. Lu, J. Shen, D. J. Crandall, and L. Shao, "Zero-shot video object segmentation via attentive graph neural networks," in *IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 9235–9244.

[113] S. Qi, W. Wang, B. Jia, J. Shen, and S.-C. Zhu, "Learning human-object interactions by graph parsing neural networks," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 407–423.

[114] J. Lu, J. Xuan, G. Zhang, and X. Luo, "Structural property-aware multilayer network embedding for latent factor analysis," *Pattern Recognition*, vol. 76, pp. 228–241, 2018.

[115] J. Bastings, I. Titov, W. Aziz, D. Marcheggiani, and K. Sima'an, "Graph convolutional encoders for syntax-aware neural machine translation," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017, pp. 1957–1967.

[116] T. H. Nguyen and R. Grishman, "Graph convolutional networks with argument-aware pooling for event detection," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2018, pp. 5900–5907.

[117] Q. Liu, H. Huang, G. Zhang, Y. Gao, J. Xuan, and J. Lu, "Semantic structure-based word embedding by incorporating concept convergence and word divergence," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2018, pp. 5261–5268.

[118] H. T. Nguyen, P. H. Duong, and E. Cambria, "Learning short-text semantic similarity with word embeddings and external knowledge sources," *Knowledge-Based Systems*, vol. 182, p. 104842, 2019.

[119] J. C. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research (JMLR)*, vol. 12, pp. 2121–2159, 2011.

[120] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, "Advances in pre-training distributed word representations," in *International Conference on Language Resources and Evaluation (LREC)*, 2018.

[121] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2007, pp. 440–447.

[122] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2004, pp. 168–177.

[123] X. Li and D. Roth, "Learning question classifiers," in *International Conference on Computational Linguistics (COLING)*, 2002.

[124] S. Riedel, L. Yao, and A. McCallum, "Modeling relations and their mentions without labeled text," in *The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML PKDD*, 2010, pp. 148–163.

[125] J. R. Finkel, T. Grenager, and C. D. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2005, pp. 363–370.

[126] D. Zeng, K. Liu, Y. Chen, and J. Zhao, "Distant supervision for relation extraction via piecewise convolutional neural networks," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015, pp. 1753–1762.

[127] A. Saraswat, K. Abhishek, and S. Kumar, "Text classification using multilingual sentence embeddings," in *Evolution in Computational Intelligence - Frontiers in Intelligent Computing: Theory and Applications (FICTA 2020)*, vol. 1176, 2020, pp. 527–536.

[128] B. Hachey, W. Radford, J. Nothman, M. Honnibal, and J. R. Curran, "Evaluating entity linking with wikipedia," *Artificial Intelligence*, vol. 194, pp. 130–150, 2013.

[129] S. Zhou, S. Rijhwani, J. Wieting, J. G. Carbonell, and G. Neubig, "Improving candidate generation for low-resource cross-lingual entity linking," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 109–124, 2020.

[130] W. Yih, M. Chang, X. He, and J. Gao, "Semantic parsing via staged query graph generation: Question answering with knowledge base," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2015, pp. 1321–1331.

[131] B. Zhang, Y. Lin, X. Pan, D. Lu, J. May, K. Knight, and H. Ji, "ELISA-EDL: A cross-lingual entity extraction, linking and localization system," in *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018, pp. 41–45.

[132] L. Gao, Z. Dai, Z. Fan, and J. Callan, "Complementing lexical retrieval with semantic residual embedding," *CoRR*, vol. abs/2004.13969, 2020.

[133] Y. Cao, L. Hou, J. Li, and Z. Liu, "Neural collective entity linking," in *International Conference on Computational Linguistics (COLING)*, 2018, pp. 675–686.

[134] O. Ganea and T. Hofmann, "Deep joint entity disambiguation with local neural attention," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017, pp. 2619–2629.

[135] P. Le and I. Titov, "Improving entity linking by modeling latent relations between mentions," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018, pp. 1595–1604.

[136] M. Xue, W. Cai, J. Su, L. Song, Y. Ge, Y. Liu, and B. Wang, "Neural collective entity linking based on recurrent random walk network learning," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2019, pp. 5327–5333.

[137] A. Sil and R. Florian, "One for all: Towards language independent named entity linking," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016, pp. 2255–2264.

[138] A. Sil, G. Kundu, R. Florian, and W. Hamza, "Neural cross-lingual entity linking," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2018, pp. 5464–5472.

[139] S. Upadhyay, N. Gupta, and D. Roth, "Joint multilingual supervision for cross-lingual entity linking," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018, pp. 2486–2495.

[140] Q. Liu, H. Huang, J. Xuan, G. Zhang, and J. Lu, "A fuzzy word similarity measure for selecting top-k similar words in query expansion," *IEEE Transactions on Fuzzy Systems*, 2020.

[141] X. Chen and C. Cardie, "Unsupervised multilingual word embeddings," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018, pp. 261–270.

[142] M. Artetxe, G. Labaka, and E. Agirre, "A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018, pp. 789–798.

[143] J. Wieting, M. Bansal, K. Gimpel, and K. Livescu, "Charagram: Embedding words and sentences via character n-grams," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016, pp. 1504–1515.

[144] S. Upadhyay, J. Kodner, and D. Roth, "Bootstrapping transliteration with constrained discovery for low-resource languages," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018, pp. 501–511.

[145] C. Tsai and D. Roth, "Learning better name translation for cross-lingual wikification," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2018, pp. 5528–5536.

[146] C. Xing, D. Wang, C. Liu, and Y. Lin, "Normalized word embedding and orthogonal transform for bilingual word translation," in *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2015, pp. 1006–1011.

[147] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with gpus," *CoRR*, vol. abs/1702.08734, 2017.

[148] A. Rosenfeld and M. Thurston, "Edge and curve detection for visual scene analysis," *IEEE Transactions on Computers*, vol. 20, no. 5, pp. 562–569, 1971.

[149] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[150] M. Hu, Y. Peng, Z. Huang, and D. Li, "Retrieve, read, rerank: Towards end-to-end multi-document reading comprehension," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019, pp. 2285–2295.

[151] S. E. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Foundations and Trends in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[152] Y. Merhav and S. Ash, "Design challenges in named entity transliteration," in *International Conference on Computational Linguistics (COLING)*, 2018, pp. 630–640.

[153] C. Dyer, V. Chahuneau, and N. A. Smith, "A simple, fast, and effective reparameterization of IBM model 2," in *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2013, pp. 644–648.

[154] X. Pan, T. Gowda, H. Ji, J. May, and S. Miller, "Cross-lingual joint entity and word embedding to improve entity linking and parallel sentence mining," in *The 2nd Workshop on Deep Learning Approaches for Low-Resource NLP, DeepLo@EMNLP-IJCNLP*, 2019, pp. 56–66.

[155] J. Daiber, M. Jakob, C. Hokamp, and P. N. Mendes, "Improving efficiency and accuracy in multilingual entity extraction," in *International Conference on Semantic Systems, ISEM*, 2013, pp. 121–124.

[156] P. Ferragina and U. Scaiella, "TAGME: on-the-fly annotation of short text fragments (by wikipedia entities)," in *Conference on Information and Knowledge Management (CIKM)*, 2010, pp. 1625–1628.

[157] M. Dubey, D. Banerjee, A. Abdelkawi, and J. Lehmann, "Lc-quad 2.0: A large dataset for complex question answering over wikidata and dbpedia," in *International Semantic Web Conference, ISWC*, 2019, pp. 69–78.

[158] S. Chen, J. Wang, F. Jiang, and C. Lin, "Improving entity linking by modeling latent entity type information," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2020, pp. 7529–7537.

[159] D. O, S. Kwon, K. Kim, and Y. Ko, "Word sense disambiguation based on word similarity calculation using word vector representation from a knowledge-based graph," in *International Conference on Computational Linguistics (COLING)*, 2018, pp. 2704–2714.

[160] F. Diaz, B. Mitra, and N. Craswell, "Query expansion with locally-trained word embeddings," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.

[161] D. Bollegala, M. Ishizuka, and Y. Matsuo, "Measuring semantic similarity between words using web search engines," in *The Web Conference (WWW)*, 2007, pp. 757–766.

[162] K. W. Church and P. Hanks, "Word association noms, mutual information, and lexicography," *Computational Linguistics*, vol. 16, no. 1, pp. 76–83, 1990.

[163] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science and Technology*, vol. 41, no. 6, pp. 391–407, 1990.

[164] P. D. Turney, "Mining the web for synonyms: PMI-IR versus LSA on TOEFL," in *European Conference on Machine Learning (ECML)*, 2001, pp. 491–502.

[165] D. Lin, "An information-theoretic definition of similarity," in *International Conference on Machine Learning (ICML)*, 1998, pp. 296–304.

[166] Y. Li, Z. Bandar, and D. McLean, "An approach for measuring semantic similarity between words using multiple information sources," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp. 871–882, 2003.

[167] D. Sánchez, M. Batet, D. Isern, and A. Valls, "Ontology-based semantic similarity: A new feature-based approach," *Expert Systems with Applications*, vol. 39, no. 9, pp. 7718–7728, 2012.

[168] A. Solé-Ribalta, D. Sánchez, M. Batet, and F. Serratosa, "Towards the estimation of feature-based semantic similarity using multiple ontologies," *Knowledge-Based Systems*, vol. 55, pp. 101–113, 2014.

[169] D. Bollegala, Y. Matsuo, and M. Ishizuka, "Measuring semantic similarity between words using web search engines," in *The Web Conference (WWW)*, 2007, pp. 757–766.

[170] M. Song, I. Song, X. Hu, and R. B. Allen, "Integration of association rules and ontologies for semantic query expansion," *Data & Knowledge Engineering*, vol. 63, no. 1, pp. 63–75, 2007.

[171] C. C. Latiri, H. Haddad, and T. Hamrouni, "Towards an effective automatic query expansion process using an association rule mining approach," *Journal of Intelligent Information Systems*, vol. 39, no. 1, pp. 209–247, 2012.

[172] A. Bouziri, C. Latiri, É. Gaussier, and Y. Belhareth, "Learning query expansion from association rules between terms," in *International Conference on Knowledge Discovery and Information Retrieval(KDIR)*, 2015, pp. 525–530.

[173] A. Bouziri, C. Latiri, and É. Gaussier, "Efficient association rules selecting for automatic query expansion," in *Computational Linguistics and Intelligent Text Processing, CICLing*, 2017, pp. 563–574.

[174] A. Abbache, F. Meziane, G. Belalem, and F. Z. Belkredim, "Arabic query expansion using wordnet and association rules," *International Journal of Intelligent Information Technologies*, vol. 12, no. 3, pp. 51–64, 2016.

[175] Q. Zhang, D. Wu, G. Zhang, and J. Lu, "Fuzzy user-interest drift detection based recommender systems," in *IEEE International Conference on Fuzzy Systems, FUZZ-IEEE*, 2016, pp. 1274–1281.

[176] H. Zuo, G. Zhang, W. Pedrycz, V. Behbood, and J. Lu, "Granular fuzzy regression domain adaptation in takagi-sugeno fuzzy models," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 2, pp. 847–858, 2018.

[177] F. Liu, G. Zhang, and J. Lu, "Heterogeneous unsupervised domain adaptation based on fuzzy feature fusion," in *IEEE International Conference on Fuzzy Systems, FUZZ-IEEE*, 2017.

[178] A. Liu, G. Zhang, and J. Lu, "Fuzzy time windowing for gradual concept drift adaptation," in *IEEE International Conference on Fuzzy Systems, FUZZ-IEEE*, 2017.

[179] Y. Song, G. Zhang, J. Lu, and H. Lu, "A fuzzy kernel c-means clustering model for handling concept drift in regression," in *IEEE International Conference on Fuzzy Systems, FUZZ-IEEE*, 2017.

[180] K. A. Crockett, N. Adel, J. O'Shea, A. Crispin, D. Chandran, and J. P. Carvalho, "Application of fuzzy semantic similarity measures to event detection within tweets," in *IEEE International Conference on Fuzzy Systems, FUZZ-IEEE*, 2017.

[181] J. Ma, G. Zhang, and J. Lu, "A method for multiple periodic factor prediction problems using complex fuzzy sets," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 1, pp. 32–45, 2012.

[182] J. I. Forcen, M. Pagola, H. Bustince, J. M. Soto-Hidalgo, and J. Chamorro-Martínez, "Adding fuzzy color information for image classification," in *IEEE International Conference on Fuzzy Systems, FUZZ-IEEE*, 2017, pp. 1–6.

[183] B. Ziólko, D. Emms, and M. Ziólko, "Fuzzy evaluations of image segmentations," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 4, pp. 1789–1799, 2018.

[184] R. Zhao and K. Mao, "Fuzzy bag-of-words model for document representation," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 2, pp. 794–804, 2018.

[185] S. Lee and J. Jiang, "Multilabel text categorization based on fuzzy relevance clustering," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 6, pp. 1457–1471, 2014.

[186] T. P. Martin, Y. Shen, and B. Azvine, "Incremental evolution of fuzzy grammar fragments to enhance instance matching and text mining," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 6, pp. 1425–1438, 2008.

[187] D. Chandran, K. A. Crockett, D. McLean, and Z. Bandar, "FAST: A fuzzy semantic sentence similarity measure," in *IEEE International Conference on Fuzzy Systems, FUZZ-IEEE*, 2013.

[188] J. Singh, M. Prasad, O. K. Prasad, M. J. Er, A. K. Saxena, and C. Lin, "A novel fuzzy logic model for pseudo-relevance feedback-based query expansion," *International Journal of Fuzzy Systems*, vol. 18, no. 6, pp. 980–989, 2016.

[189] J. Singh and A. Sharan, "A new fuzzy logic-based query expansion model for efficient information retrieval using relevance feedback approach," *Neural Computing and Applications*, vol. 28, no. 9, pp. 2557–2580, 2017.

[190] Q. Liu, H. Huang, J. Lu, Y. Gao, and G. Zhang, "Enhanced word embedding similarity measures using fuzzy rules for query expansion," in *IEEE International Conference on Fuzzy Systems, FUZZ-IEEE*, 2017, pp. 1–6.

[191] Y. Gupta and A. Saini, "A novel fuzzy-pso term weighting automatic query expansion approach using combined semantic filtering," *Knowledge-Based Systems*, vol. 136, pp. 97–120, 2017.

[192] O. Levy, Y. Goldberg, and I. Dagan, "Improving distributional similarity with lessons learned from word embeddings," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 211–225, 2015.

[193] G. Piatetsky-Shapiro, "Discovery, analysis, and presentation of strong rules," in *Knowledge Discovery in Databases*. AAAI/MIT Press, 1991, pp. 229–248.

[194] G. Cao, J. Nie, J. Gao, and S. Robertson, "Selecting good expansion terms for pseudo-relevance feedback," in *Annual International ACM SIGIR Conference*

on Research and Development in Information Retrieval (SIGIR), 2008, pp. 243–250.

[195] C. Lee, "Fuzzy logic in control systems: fuzzy logic controller. I," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 2, pp. 404–418, 1990.

[196] W. Li, D. Ganguly, and G. J. F. Jones, "Using wordnet for query expansion: ADAPT @ FIRE 2016 microblog track," in *Forum for Information Retrieval Evaluation*, 2016, pp. 62–65.

[197] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781*, 2013.

[198] S. E. Robertson, H. Zaragoza, and M. J. Taylor, "Simple BM25 extension to multiple weighted fields," in *Conference on Information and Knowledge Management (CIKM)*, 2004, pp. 42–49.

[199] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, 2008.

APPENDIX

Table 1: Abbreviations and their explanations.

Abbreviation	**Explanation**
NLP	natural language processing
IR	information retrieval
KB	knowledge base
SENSE	semantic structure-based word embedding
GCN	graph convolution network
LSA	latent semantic analysis
LDA	latent Dirichlet allocation
GloVe	global vectors model
NN	network network
NNLM	neural network language model
CBOW	continuous bag of words
PTM	pre-trained language model
ELMo	embedding from language models
GPT	generative pre-trained transformer
BERT	bidirectional encoder representations form transformers
ERNIE	enhanced language representation with informative entities
POS	part-of-speech
K-BERT	knowledge-enabled language representation model
KEPLER	knowledge embedding and pre-trained language representation
DA	domain adaptation

HDA	heterogeneous domain adaptation
SCL	structural correspondence learning
MI	mutual information
SFA	spectral feature alignment
PLSR	partial least squares regression
TWE	topical word embedding
QA	question answering
OOV	out-of-vocabulary
SGD	stochastic gradient descent
CNN	convolutional neural network
NYT	New York Times corpus
PCNNs	Piecewise Convolution Netural Networks
XEL	cross-lingual entity linking
LRL	low-resource language
HRL	high-resource language
CSLS	cross-domain similarity local scaling metric
NMS	non-maximum suppression
KBQA	question answering over knowledge base
CR	candidate retrieval
ED	entity disambiguation
SVD	singular value decomposition
FWS	fuzzy word similarity
XML	Extensible Markup Language
MAP	mean average precision
LM	language model
NP	noun phrases