UNIVERSITY OF LIVERPOOL

Joint Approaches for Learning Word Representations from Text Corpora and Knowledge Bases

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor in Philosophy by

Mohammed Alsuhaibani

March 2020

Dedication

To my father, you are in a better place now.

Acknowledgements

Alhamdulillah, praise be to Allah Almighty for His graces and guidance.

With delight and gratefulness, I would like to acknowledge all who supported and encouraged me during my PhD journey. This thesis would not have been possible without their support.

First and foremost, I would like to express my profound gratitude and appreciations to my primary supervisor Professor. Danushka Bollegala, for his continuous support and encouragement. Without his guidance, this PhD would not have been achievable. He spared no effort to share his extensive scientific knowledge to help me throughout the journey of my PhD. I have been privileged and pleased to work under his supervision.

I would also like to thank my secondary supervisor Professor. Yannis Goulermas and my advisors, Professor. Frans Coenen, Professor. Frans Oliehoek and Professor. Katie Atkinson for their constructive feedback and suggestions. Special thanks are also due to my friends and colleagues in the Natural Language Processing (NLP@Liv) group at the Department of Computer Science for all the helpful discussions we had throughout the years and the kind support and assistance whenever needed.

I am out of words to express my deepest gratitude to my family. To my parents for the unfailing love and spiritual support. To my wife, Futoon, for the endless support and standing beside me all these years. To my children, Jenan and Abdulrahman, for the gift they are in my life. To all my brothers and sisters for the constant moral support.

Finally, I would also like to extend my gratitude to Qassim University and Saudi Cultural Bureau in London for their generous sponsorship to my postgraduate studies.

Abstract

The work presented in this thesis is directed at investigating the possibility of combining text corpora and Knowledge Bases (KBs) for learning word representations. More specifically, the aim was to propose joint approaches that leverage the two types of resources for the purpose of enhancing the word meaning representations. The main research question to be answered was "*Is it possible to enhance the word representations by jointly incorporating text corpora and KBs into the word representations learning process? If so, what are the aspects of word meaning that can be enhanced by combining those two types of resources?*".

The primary contribution of the thesis is three main joint approaches for learning word representations: (i) Joint Representation Learning for Additional Evidence (JointReps), (ii) Joint Hierarchical Word Representation (HWR) and (iii) Sense-Aware Word Representations (SAWR). The JointReps was founded to improve the overall semantic representation of words. To this end, it sought additional evidence from a KB to the co-occurrence statistics in the corpus. In particular, JointReps enforced two words that are in a particular semantic relationship in the KB to have similar word representations. The HWR approach was then proposed to learn word representations in a specific order to encode the hierarchical information in a KB in the learnt representations. The HWR considered not only the hypernym relations that exist between words in a KB, but also contextual information in a text corpus. Specifically, given a training corpus and a KB, HWR learnt word representations that simultaneously encoded the hierarchical structure in the KB as well as the co-occurrence statistics between pairs of words in the corpus. A particularly novel aspect of the HWR approach was that it exploits the full hierarchical path of words existing in the KB. The SAWR approach was then introduced to consider not only word representations but also the different senses (different meanings) associated with each word. The SAWR required the learnt representations to predict the word and the senses accurately. It learnt the sense-aware word representations jointly using both unlabelled and sense-labelled text corpora.

The approaches were comprehensively analysed and evaluated in various standard and newly-proposed tasks using a wide range of benchmark datasets. The evaluation was conducted to compare the quality of the learnt word representations by the proposed approaches with word representations learnt by sole-resource baselines and previously proposed joint approaches in the literature. All the proposed joint approaches have proven to be effective for enhancing the learnt word representations. More specifically, the proposed joint approaches were found to report significant improvements over the approaches that use only one type of resources and the previously proposed joint approaches.

Contents

List of Figures

List of Tables

Notations

The following list describes several abbreviations and notations that will be used throughout the thesis:

AI	Artificial Intelligence
AUC	Area Under Curve
C-NET	Categorical Knowledge
CCA	Canonical Correlation Analysis
CL	Computational Linguistics
COALS	Correlated Occurrence Analogue to Lexical Semantic
COMP	Compositional
CR	Customer Reviews dataset
DH	Direct Hypernym
DWNN	Dynamic Weighting Neural Network
F1	F-Score
GloVe	Global Vectors
HAL	Hyperspace Analogue to Language
HNE	Hedged Nearest Neighbour Expansion
HWR	Hierarchical Word Representation
IR	Information Retrieval
JointReps	Joint Word Representations Learning For Additional Evidence
K-NN	K-Nearest Neighbour
KB	Knowledge Base
LEAR	Lexical Entailment Attract-Repel
LR-MVL	Low-Rank Multi-View Learning
LSA	Latent Semantic Analysis
MC	Miller-Charles dataset
MEN	Marco, Elia and Nam dataset
ML	Machine Learning
MR	Movie Reviews dataset
MSSG	Multi-Sense Skipgram
NCE	Noise-Contrastive Estimation

NEG	Negative Sampling
NER	Name Entity Recognition
NLP	Natural Language Processing
NLU	Natural Language Understanding
NNE	Nearest Neighbour Expansion
NNLM	Neural Network Language Model
NP-MSSG	Non-Parametric Multi-Sense Skipgram
PCA	Principle Component Analysis
PMI	Pointwise Mutual Information
POS	Part-Of-Speech
PPDB	Paraphrase Database
PPMI	Positive Pointwise Mutual Information
R-NET	Relational Knowledge
RC-NET	Relational and Categorical Knowledge
RCM	Relation Constraint Model
Retro	Retrofitting model
RG	Rubenstein-Goodenough dataset
ROC	Receiver Operating Characteristic
RW	Rare Words dataset
SAWR	Sense-Aware Word Representations
SCWS	Stanford's Contextual Word Similarities dataset
Sem	Semantic Analogy dataset
SenseEmbed	Sense Embeddings
SenseRetro	Sense Retrofitting
SGD	Stochastic Gradient Descent
SimLex	SimLex-999 dataset
SKB	Static Knowledge Base
SUBJ	Subjectivity dataset
SVD	Singular Value Decomposition
Syn	Syntactic Analogy dataset
TR	Stanford Sentiment Treebank dataset
VSM	Vector Space Model
WS	WordSim353 dataset
WSD	Word Sense Disambiguation

Chapter 1

Introduction

1.1 Overview

Understanding the meaning of textual data is essential for Natural Language Processing (NLP) systems. As individual lexical items lie at the core of the natural languages (i.e. English, Spanish, Arabic etc.), understanding the meaning of words is, therefore, a crucial aspect of NLP systems that aim to process natural languages. To enable NLP systems to understand the meaning of words, we must develop systems that can represent textual data in a form that allows computers to effectively read and understand such data. Thus, research on learning to represent the meaning of words using linear algebraic structures such as vectors has emerged as a fundamental task in NLP [97, 138]. As such, word meaning/semantic representations (commonly known as word representations) can be defined as linear algebraic structures, such as vectors, associated with individual words that hold informative features about the meaning of each particular word [136].

Learning word representations that accurately represent the meaning of words has been shown to play a significant role in improving the performance of numerous NLP and Machine Learning (ML) tasks such as word similarity measurement [136], sentiment analysis [147], machine translation [150], name entity recognition (NER) [83], document classification [82], relation extraction [113], word sense disambiguation (WSD) [74], question answering [25] and textual entailment [35], among many others. Moreover, accurately representing the meanings of individual words can be used to build semantic representations for larger lexical units such as sentences [78] or documents [86] in a bottom-up fashion by applying semantic compositional operators on the word-level semantic representations [12]. Consequently, creating accurate word-level semantic representations is a core task in NLP.

Over the years, two types of resources have been frequently used for learning word representations: (i) text corpora, such as ukWaC [54], COCA [38] and Gigaword [118], which are often defined as a collection of naturally occurring language text, either written or a transcribed speech, collected from a wide range of sources and chosen to characterise a state or variety of a language [96, 131], and (ii) manually created lexical Knowledge Bases (KBs) such as WordNet [104], FrameNet [10] and the Paraphrase Database (PPDB) [58], which are commonly known as repositories of computational information about lexical concepts and their relationships, intended to be useful in many fields including computational linguistics (CL) and artificial intelligence (AI) [6].

By dint of the distributional hypothesis: “words that occur in the same contexts tend to have similar meanings” [65] which was popularised by Firth’s famous quote “you shall know a word by the company it keeps” [56], text corpora have been used successfully to learn distributional representations of words. In such distributional word representations, the semantic representation of a word can be provided by a vector. The dimensions of such a vector correspond to the contextual words that co-occur with the target word in contexts such as a window of tokens, a sentence or a document. Specifically, the context in which a target word, w, co-occurs with other words provides useful information about the semantic meaning of w. For example, if we were told that w is a *food*, w is made with *flatbread*, w is commonly *topped with tomato and cheese* and is *a dish of Italian origin*, we might guess that w is a *pizza*. The words that co-occur with the target word w such as *food*, *flatbread*, *topped with tomato and cheese* and *dish of Italian origin* represent various semantic attributes related to w. This means that words which tend to occur in similar contexts are likely to have similar meanings, and thus, similar distributional representations. Therefore, given a sufficiently large text corpus, we can learn the distributional word representation for a particular word using the contextual words that co-occur with it in a given corpus.

Learning word representations purely from unlabelled large text corpora is attractive because such corpora are often easy to obtain and obviate the need for any manual data annotation. However, corpus-based word representations are learnt solely from large text corpora, ignoring the knowledge available in KBs. Consequently, there is no guarantee that the word representations learnt by the unsupervised, corpus-based, approaches will accurately capture the semantic relations that exist between words. Indeed, concerns about the ability of corpus-based word representation learning methods to accurately estimate the strength of the semantic relationships that exist among words [51, 108, 145, 148] or to capture complex linguistic phenomena such as task- or domain-specificity [7, 23, 29, 111] and ambiguity [73, 76, 77, 126], have been raised in the NLP community.

On the other hand, manually created KBs, such as FrameNet [10] and WordNet [104], provide an alternative method for representing the meanings of words. A word in such KBs is defined using its linguistic properties and the semantic relations it has with other words. For example, in WordNet, the word *dark* is defined as a synonym of *aphotic*, an antonym of *light*, a hyponym of *night* and has a hypernym *illumination*. Its Part-Of-Speech (POS) can be a *noun* or an *adjective*, and it has *six different senses* such as *unilluminated area* and *unenlightened state*, etc. The semantic relations and linguistic properties of *dark* found in the KB provide invaluable information about the meaning of *dark*. Thus, it can be represented in a vector in which the dimensions correspond to its semantic and linguistic properties [52].

Although KBs provide valuable information about the meaning and relationships of words, such information is manually curated by human experts and is thus costly and time-consuming to produce. Additionally, most KBs are manually updated on a periodical basis, and consequently new words or new uses of existing words (neologisms) tend not to be well covered by the manually constructed KBs. Moreover, in practice, KBs tend to be much smaller than text corpora in size [4]; therefore, each KB contains a limited number of entries for a particular word.

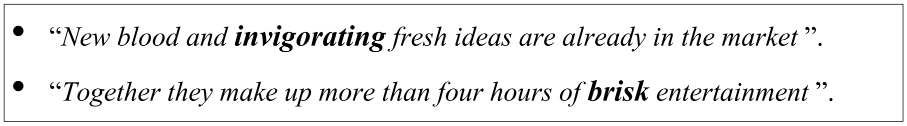
From the preceding discussion, it is evident that using a single type of resource, either text corpora or KBs, as the only source for learning word representations results in various limitations. However, as will be demonstrated in detail in the next section, combining the corpora and KBs offers complementary strengths when it comes to learning word representations, which is what the work presented in this thesis is motivated by. More specifically, **the fundamental idea presented in this thesis is to investigate and explore joint approaches that combine**

the two types of resources to enhance the overall process of learning word representations.

1.2 Motivations

As described above, the primary motivation for the work presented in this thesis is the complementary strengths of text corpora and KBs when it comes to learning word representations. With the help of concrete examples (sentences extracted from text corpora), several scenarios in which these complementary strengths are clearly prominent, are highlighted below:

Additional Evidence. As noted earlier, distributional word representations are learnt from surface-level word co-occurrences in a corpus, where words sharing common contexts share similar meanings and thus have similar distributional representations. For distributional techniques to accurately represent the meaning of words, they require an abundance of occurrences for each word in the corpus. However, this statistical requirement can be problematic when some words occur rarely in the corpus, which is, in fact, the case even with massively large corpora [121]. That is, the corpus might not be sufficiently large to obtain reliable word co-occurrence counts. Therefore, there is a need for additional evidence to support the rare co-occurrence. To demonstrate the need for such additional evidence, let us consider the two sentences below:



- "*New blood and* ***invigorating*** *fresh ideas are already in the market* ".
- "*Together they make up more than four hours of* ***brisk*** *entertainment* ".

The two words, *invigorating* and *brisk*, are considered rare words in English [94] and they are recorded as synonyms in WordNet. Because they are rare words, a corpus will consequently have few contexts in which they appear. Let us assume that these two sentences are the only sentences in the corpus where *invigorating* and *brisk* occur. We can see that the two sentences have no common words among them. Therefore, if we were to use the corpus alone to learn the word representations, it would be difficult to accurately estimate the strength of the relationship between *invigorating* and *brisk*. This would eventually result in learning inadequate representations. Consequently, there is a need for additional evidence that can strengthen the similarity between those rare words since their co-occurrence in the corpus is not sufficient. This is already available in the KB where *invigorating* and *brisk* are recorded as synonyms. Similarly, when using the corpus alone, the lack of sufficient context for *invigorating* and *brisk* might also inappropriately result in words such as *blood*, *fresh* and *market* having similar representations as *invigorating*, because it is the only sentence where those words co-occur. Nevertheless, a KB would overcome this problem by providing that *invigorating* is a synonym of words such as *brisk*, *animating* and *energetic* but not *blood*, *fresh* or *market*.

Another situation which calls for additional evidence is when related, but dissimilar, words share similar contexts. For example, antonyms are likely to occur in similar contexts [30, 112]. Therefore, in such cases, solely using the distributional information to learn word representations can be problematic. To illustrate the challenge here, let us take a look at the two antonyms, *cold* and *hot*, given in the two sentences below:

- "*Those areas, especially since climate change, have more reliable* ***cold*** *weather than Germany or Alsace*".
- "*Climate change does not imply* ***hot*** *weather and droughts, it implies extreme weather* ".

We can see that the two words appear in similar contexts, as the two sentences have multiple words in common. Consequently, using only contextual words to learn the word representations can result in words with contrasting meanings being given inappropriately similar representations. However, various other contexts would also be observed where *cold* appears in similar contexts to its synonyms such as *frigid* and *icy*. These synonyms can be easily obtained from a KB, and subsequently used as additional evidence alongside the information provided by the distributional co-occurrence. As such, we can further pull *cold* closer to its synonyms (e.g. *frigid* and *icy*) but not to its antonyms (e.g. *hot* and *warm*).

Fine-Tuned Word Representations. The complementary strengths of a corpus and a KB is also apparent when the learnt word representations are desired to be fine-tuned to encode a particular semantic relation to suit a task of interest. For example, prior work has shown that distributional representations of words are capable, to an extent, of encoding hierarchical information (hypernym relations) existing among words, due to the lexical patterns available in the corpus [101, 119, 128], thereby helping some NLP tasks such as recognising textual entailment [35] and text generation [19]. However, the lexical patterns extracted from a corpus could be insufficient and might lead to incorrect inferences. To demonstrate the potential issue here, we will consider the two sentences below:

- *“**Carnivores such as cheetah** have become almost extinct in the region”.*
- *“Some birds recorded in **Africa such as gadwall** ”.*

In the first sentence, matching the pattern *X such as Y* in “*carnivores such as cheetah*” does indeed express a hypernym relation between *carnivore* and *cheetah*. However, in the second sentence matching the same pattern in “*Africa such as gadwall*” will incorrectly detect *Africa* and *gadwall* as having a hypernym relation. Such noise in distributional approaches can be reduced by incorporating a KB, which will explicitly state the hierarchical relations between words, and thus help to fine-tune the learnt representations to reflect the hierarchy.

Word Sense Disambiguation. By using both a corpus and a KB we can overcome the ambiguity of a word. In fact, one of the limitations associated with learning distributional representations of words solely from text corpora is that they only represent each word by a single vector [71]. As such, the potentially multiple senses of a word are ignored. To illustrate how combining the two types of resources might help to overcome this ambiguity, let us consider the two sentences above. The word *bank* in the first sentence refers to the *financial institution*, while the *bank* in the second sentence refers to the *shore of a river*. The two senses

- “*The **bank** plans to pay out between 40-60% of their profit* ”.
- “*He ran along the **bank** of the river before he jumped into the water* ”.

of *bank* are significantly different, and thus having a single representation for such an ambiguous word is inadequate. However, in KBs, words are grouped with other words that share similar senses, and therefore can be used to manually or automatically disambiguate the corpus, and thus learn sense-aware word representations.

From the foregoing, it is clear that taking the advantages of each resource complements the other. Therefore, the work presented in this thesis seeks to explore joint approaches for learning word representations that utilise both text corpora and KBs. More specifically, this thesis explores joint approaches that are capable of addressing each deficiency using a single resource and consequently enhancing the learnt word representations. To this end, several methods to extract information from KBs have been proposed in this thesis to be incorporated with the distributional information obtained from a corpus.

1.3 Research Questions

Given the above motivations, the research question which this thesis seeks to answer is as follows: ***Is it possible to enhance word representations by jointly incorporating text corpora and KBs into the word representation learning process? If so, what are the aspects of word meaning that can be enhanced by combining those two types of resources?***

Answering the above research question requires the resolution of the following subsidiary questions:

1. What is the most appropriate mechanism to incorporate a KB with a corpus to provide additional evidence to the distributional information obtained from the corpus?
    - How can a KB provide additional evidence to accurately estimate the relationship between rare words, and to push the words to their semantically similar words?
2. Given a solution to (1), can we utilise the corpus co-occurrence statistics to compensate for the limited number of entries for the words in a KB?
    - If so, what is the appropriate mechanism to use the corpus co-occurrence information to expand a KB?

3. How can a KB and a corpus be best combined to fine-tune word representations to a target task or to represent a particular semantic relation?

   - What is the best mechanism to extract KB and corpus data for that purpose?

4. Given solutions to (3), how can we properly evaluate the enhancement brought by the joint approach on the learnt word representations?

5. How can sense-aware word representations best be learnt from sense related information available in a KB and contextual clues in the corpus?

1.4 Contributions

The primary goal of this thesis is to explore joint approaches for learning word representations from text corpora and KBs. To achieve that goal, this thesis presents three main joint approaches (illustrated in Figure 1.1) demonstrating that it is indeed possible to enhance the word representations by incorporating the two type of resources. These joint approaches have led to a number of contributions with respect to the NLP community, and can be summarised as follows:

1. A joint approach that combines a KB into the learning process with a corpus to provide additional evidence. The KB's knowledge is incorporated as relational constraints that must be satisfied by the learnt word representations. This work has been published as a conference paper at the Association for the Advancement of Artificial Intelligence (AAAI) conference, 2016 [21].

2. Two approaches that expand the KB from the corpus co-occurrence statistics proposed as an enhancement to the approach given in (1) above. This work has been published as a journal paper at the Public Library of Science (PLOS ONE) journal, 2018 [4].

3. A joint approach that fine-tunes the word representations to encode the hierarchical structure between words, using the hypernym relations that exist between words in a KB and the contextual information in a corpus. This work has been published as a conference paper at the Automated Knowledge Base Construction (AKBC) conference, 2019 [5].

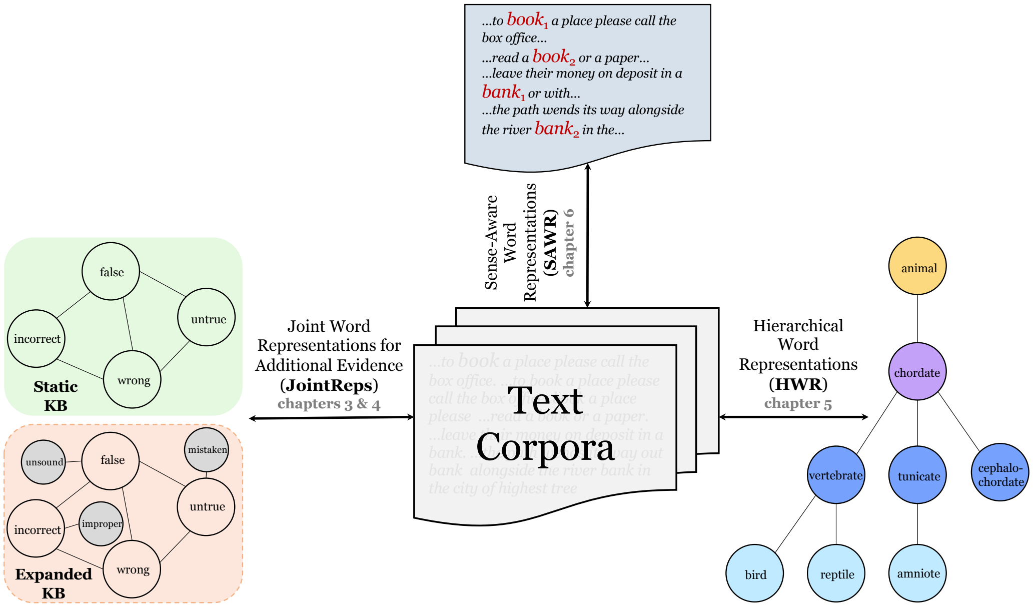

...to book$_1$ a place please call the box office...
...read a book$_2$ or a paper...
...leave their money on deposit in a bank$_1$ or with...
...the path wends its way alongside the river bank$_2$ in the...
Sense-Aware Word Representations (SAWR) chapter 6
false
untrue
incorrect
wrong
Static KB
Joint Word Representations for Additional Evidence (JointReps) chapters 3 & 4
Text Corpora
Hierarchical Word Representations (HWR) chapter 5
animal
chordate
vertebrate
tunicate
cephalo-chordate
bird
reptile
amniote
unsound
false
mistaken
untrue
incorrect
improper
wrong
Expanded KB

Figure 1.1: Illustration of the joint approaches proposed in this thesis.

4. An evaluation task with a benchmark dataset (publicly available) to evaluate any fine-tuned word representations for hierarchical information. This work has been published as a conference paper at the Automated Knowledge Base Construction (AKBC) conference, 2019 [5].

5. An evaluation task to understand the compositional structure between words and their hypernyms. This work is currently under review for the Computational Linguistics Journal, 2019.

6. A joint approach that learns sense-aware word representations from unlabelled and sense-labelled (KB's linked senses) corpora. This work has been published as a conference paper at the Language Resources and Evaluation (LREC) conference, 2018 [3].

It is worth mentioning here that in all the joint word representation learning approaches proposed in this thesis, ukWaC has been used as a corpus. However, none of the proposed methods is restricted to using only ukWaC or any other particular corpus. Any corpus, such as Gigaword [118], Google News [99], and American National Corpus [75], to name a few, where the contextual co-occurrence statistics between words are attainable, can be used with the proposed methods. The ukWaC was selected as it is one of the largest publicly available text corpora and has been widely used in prior related work [49, 85, 89, 98]. Moreover, Baroni et al. [13] have conducted comprehensive experiments on various word representations learning methods using ukWaC, which shows consistency in the performance of all the methods. As a KB, the proposed joint approaches used WordNet as a lexical KB. WordNet has been successfully used in prior work that aims to jointly learn word representations [51, 108, 145, 148]. Nevertheless, any KB that specifies the relationships that exist between words could be used as KB with all the proposed methods, such as FrameNet and PPDB. In particular, we do not assume any structural properties unique to a specific KB.

1.5 Published Work

The main contributions of this thesis have already been published in relevant peer-reviewed conferences and journals as follows:

Journal Papers

1. Mohammed Alsuhaibani, Danushka Bollegala, Takanori Maehara, and Kenichi Kawarabayashi: *Jointly Learning Word Embeddings Using a Corpus and a Knowledge Base*, PLoS ONE, pp. 1-26, Vol. 13, no. 3, 2018. **Chapter 4**.

2. Mohammed Alsuhaibani, Takanori Maehara and Danushka Bollegala: *HWE: Hierarchical Word Embeddings.* Computational Linguistics, July 2019 [under review]. **Chapter 5**.

Conference Papers

1. Danushka Bollegala, Mohammed Alsuhaibani, Takanori Maehara, and Kenichi Kawarabayashi: *Joint Word Representation Learning using a Corpus and a Semantic Lexicon*, 30th AAAI Conference on Aritificial Intelligence (AAAI), Arizona, USA, 2016. **Chapter 3**.

2. Mohammed Alsuhaibani and Danushka Bollegala: *Joint Learning of Sense and Word Embeddings*, in Proceeding of the 11th edition of the Language Resources and Evaluation Conference, Miyazaki, Japan, 2018. **Chapter 6**.

3. Mohammed Alsuhaibani, Takanori Maehara and Danushka Bollegala: *Joint Learning of Hierarchical Word Embeddings from a Corpus and a Taxonomy*, in Proceeding of the 1st Automated Knowledge Base Construction Conference (AKBC), Amherst, Massachusetts, USA, 2019. **Chapter 5**.

1.6 Thesis Outline

The remainder of the thesis is organised as follows:

Chapter 2: Background and Related Work. This chapter presents background information related to learning word representations and a literature review of the related research to the work presented in this thesis. The chapter commences with an overview of word vector representations and their emergence as a key player in the NLP field. This is followed by a review of the relevant previous work with respect to the word representations learning. The relevant work is divided into three main parts depending on the resource used in the learning process: corpus-based, KB-based and joint approaches.

Chapter 3: Joint Representation Learning for Additional Evidence. This chapter presents a novel method called *Joint Representation Learning for Additional Evidence* (JointReps). This is the first of three main joint approaches proposed in this thesis for learning word representations using a corpus and a KB. In this approach, the KB's knowledge is used as additional evidence alongside the information provided by the distributional co-occurrence. Specifically, JointReps simultaneously predicts the co-occurrences of two words in a sentence subject to the relational constrains given by a KB. The relations that exist between words in the KB have been used to regularise the word representations learnt from the corpus.

Chapter 4: Dynamic Knowledge Base Expansion. In this chapter, the incorporation process of the KB in the JointReps is enhanced by proposing two novel expansion methods: Nearest Neighbour Expansion (NNE) and Hedged Nearest Neighbour Expansion (HNE) that expand the KB using the information extracted from the corpus. In both methods, the KB is expanded and dynamically updated by extracting features from corpus based on co-occurrence counts, instead of considering only the original information in the KB.

Chapter 5: Fine-Tuned Word Representation for Hierarchical Information. This chapter looks into fine-tuning the word representations to represent the hierarchical structure that exists among words and presents the second joint approach proposed in this thesis. Specifically, a method called the *Joint Hierarchical Word Representation* (HWR) is proposed, in a specific order to encode the hierarchical structure of a KB in a vector space. To learn the hierarchical word representations, HWR method considers not only the hypernym relations that exist between words in a KB, but also the contextual information in a text corpus.

Chapter 6: Sense-Aware Word Representations. This chapter intends to address the polysemy and the corpus ambiguity problems and presents the third joint approach proposed in this thesis. Specifically, it presents a *Sense-Aware Word Representations* (SAWR) approach that jointly learns from both unlabelled and sense-tagged (where the senses are tagged using a KB) corpora. The proposed SAWR can learn both word and sense representations in the same vector space by efficiently exploiting both types of resources.

Chapter 7: Conclusion. In this chapter, the thesis is concluded by summarising the main findings regarding the research question and the associated subsidiary questions. The chapter then provides some discussions about the possible direction for future work.

Chapter 2

Background and Related Work

2.1 Introduction

As discussed in Chapter 1, the work conducted in this thesis seeks to explore joint approaches for learning word representations from text corpora and Knowledge Bases (KBs). This chapter will thus begin (Section 2.2) by providing the necessary background concerning representation learning in general (Section 2.2.1) and word representation learning in particular (Section 2.2.2). That is followed by Section 2.3, in which a comprehensive review of the work related to this thesis is presented. The section commences with a review of corpus-based approaches in which word representations are learnt purely from text corpora (Section 2.3.1). The corpus-based approaches are categorised into: (i) count-based, where the words are represented directly from the co-occurrence statistics, (ii) prediction-based, in which the word representations are learnt to predict the co-occurrences, and (iii) hybrid approaches of count- and prediction-based. The following section (Section 2.3.2) then provides a discussion on approaches for learning word representations solely from KBs, which are referred to as KB-based approaches. Next, in Section 2.3.3, a more focused discussion on the joint learning approaches, which is the closest line to the work presented in this thesis, is provided. The section considers joint approaches for: (i) additional evidence from the KB, (ii) fine-tuning the word representations for a specific relation, and (iii) sense representations. The chapter is then wrapped-up in Section 2.4 with a summary of the material presented in the chapter.

2.2 Background

The necessary background to the work presented in this thesis is provided in this section. It commences, Section 2.2.1, with an overview of representation learning in general and how data representation plays a major role in NLP and ML. Next, Section 2.2.2 presents word representation learning process and shows how the NLP, lexical semantic field in particular, used representation learning techniques to capture word meaning and learn word representations.

2.2.1 Representation Learning

The ease (or difficulty) of many information processing tasks relies heavily on how the information is represented. This is a general principle that is applicable to almost everything, including matters of daily life, computer science in general and ML. For example, a person might find it a straightforward task to divide 225 by 9 using long division. However, the task would become considerably less straightforward if the numbers are represented using Roman numerals. That is, if the same person is asked to divide CCXXV by IX, it is more likely that he/she would first convert the numbers to an Arabic numeral representation to permit the long division procedure [61].

In the same vein, the performance of ML algorithms is substantially dependent on the choice of data representation. For that reason, data representation (also referred to as feature engineering) often consumes the greatest effort during the development of machine learning models. It involves the design of pre-processing pipelines and the reconstruction of data to generate representations in terms of features that capture inherent aspects within the data [16]. A good representation must be able to extract all the essential information about the data. However, obtaining a good representation has always been a challenging task.

Until recently, most of the feature engineering efforts in machine learning models were performed by human domain experts. That is because feature engineering is essentially the process of taking advantage of human intuition and prior knowledge to find the expressive features of a task [16]. However, its high dependence on manual effort makes it costly. Consequently, minimising the manual effort of feature engineering would be ideal for expanding the scope and promoting the applicability of ML, and more importantly, for advancing toward AI. Representation learning has thus emerged as an alternative approach that eliminates the excessive reliance on manual efforts to represent data.

Representation learning is the process of *automatically* learning a representation of data in terms of features that capture all the relevant and necessary information about the data [15]. Several research areas, including vision recognition [18, 32, 68, 80], speech recognition [36, 37, 40, 107, 130], signal processing [26, 63, 64, 146], and NLP [17, 24, 34, 59, 67, 100, 132] have been among the prime beneficiaries of representation learning. For NLP in particular, there have since been numerous efforts to use representation learning. More specifically, lexical semantics (the study of word meaning) is one particular field of NLP that has received a great deal of recent interest. There in, applying the representation learning methods to capture word meaning is widely known as word representation learning [76].

2.2.2 Word Representation Learning

Applying the techniques of representation learning for lexical semantics leads to word representations [50]. Word representations (alternatively referred to as word semantics, word meaning representations, or word embeddings) are linear algebraic structures, often vectors, associated with individual words that hold informative features about the meaning of the word [136]. Thus, given a set of words, which are fundamental units of natural languages, the aim here is to automatically learn representations for those words. Learning word representations that accurately represent the meaning of words is of significant importance in numerous NLP tasks such as word similarity measurement, sentiment analysis, machine translation, document classification and relation extraction [82, 136, 147, 150], to name a few.

The distributional hypothesis has traditionally been the fundamental pillar in learning word representations in NLP. It suggests that the meaning of a word can often be guessed from the contexts it appears within. Thus, we can represent a word meaning from its distributional information in an observed context. The roots of this idea can be traced back to the 1950s when Wittgenstein [143] wrote “the meaning of a word is its use in the language”, which was then formulated by the famous quotes of Harris [65] and Firth [56], respectively, as “the complete meaning of a word is always contextual” and “you shall know a word by the company it keeps". Distributional models then put the idea of representing the meaning as distribution into practice by automatically representing words using the contexts in which they were observed in a text corpus [48]. The context here could be a window of tokens, a sentence or a document. In the simplest form, it is the window of tokens surrounding the target words (e.g. five words to the left and five words to the right), which is what we refer to as a context henceforth in this thesis.

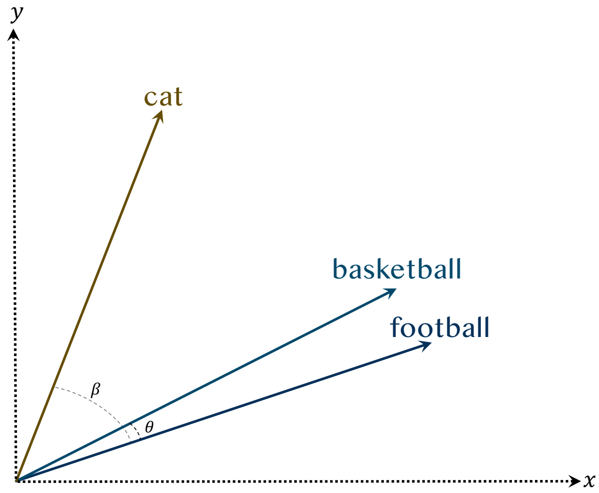


Figure 2.1: Example of computing the similarity between words as the *cosine* of the angle between their word vectors in two-dimensional vector space.

The Vector Space Model (VSM) is the backbone of the distributional models, in which a word is represented by a point in a high-dimensional space (a vector in a vector space). The dimensions correspond to the contextual words that co-occur with that target word within a context window, and the coordinates correspond to the co-occurrence counts. Points that are close together in the space are semantically similar, whereas points that are far apart are semantically dissimilar [138]. This means that words that are distributionally similar will reside close together in the vector space. The similarity between words might then be measured by their proximity in the vector space. One straightforward way of measuring the similarity between words is by using the *cosine* similarity between word vectors, which computes the *cosine* of the angle between them [50]. Figure 2.1 shows an example of three word vectors in a two-dimensional vector space. The $cosine(\theta)$ between the two semantically similar words, *football* and *basketball*, is greater than the $cosine(\beta)$ of the semantically dissimilar words, *football* (or *basketball*) and *cat*. That is, the smaller the angle between the vectors, the higher the similarity between the words.

The construction of a vector space of word meaning begins with extracting word co-occurrence statistics from large unlabelled text corpora. Such text corpora are available for many of the world's prominent natural languages; hence the unsupervised learning of distributional word representations has been vastly popular. To illustrate how VSM is constructed from the distributional information, let us consider the toy sample corpus given in Figure 2.2.

many are brought up with	***cat***	*and small breed dog so that*
his other favourite sport,	***basketball***	*when weather at the city was*
in England, they know this	***football***	*and how to win this league*
mammal that is similar to a	***dog***	*or wolf, but being marsupials*
still unknown if an infected	***cat***	*or other pet could pass influenza*
the sport of the day might be	***football***	*but everyone else is running*
provides a local high school	***football***	*player for game tickets each week*
he told him that for a large	***dog***	*to make a practical pet, you need*
you decided yet which sport,	***basketball***	*or football, you will focus on in*
ragdolls, a supercute breed of	***cat***	*plus more than 100 cats of all*
not surely a pure breed	***dog***	*the little dog continues to pad*
it was a game in a recreational	***basketball***	*league on a recent Saturday*
more than any other sport,	***football***	*is primarily a television show*
a player can take an elbow in	***basketball***	*or get hit with a fastball in*
which is called a civet	***cat***	*the small, furry mammal with big*

Figure 2.2: Sample sentences extracted from the ukWaC [54] corpus, with a focus on the four target words that we want to obtain the representations for.

Assume that we are interested in learning the representations of the four target words *cat*, *dog*, *football* and *basketball*. The context window here is the five words preceding and succeeding these target words. For each target word, we extract the co-occurrence counts within the context window. Table 2.1 shows a selection from the word-word co-occurrence matrix. The rows of the matrix given in Table 2.1 are the representations of the target words, whereas the columns (the selected context words) are the features of those word representations. We can see that the semantically similar words, *cat* and *dog*, have similar vector representations

	pet	*sport*	*breed*	*league*	*mammal*	*player*
cat	1	0	2	0	1	0
dog	1	0	3	0	2	0
football	0	2	0	2	0	1
basketball	0	1	0	2	0	1

Table 2.1: Co-occurrence matrix of four target words (four word vector representations) with selected context words computed from the toy sample corpus given in Figure 2.2.

(i.e. $cosine(\overrightarrow{cat}, \overrightarrow{dog}) = 0.92$) because they tend to occur with similar context words such as *pet*, *mammal* and *breed*. Likewise, the vector representations of *football* and *basketball* are more similar to each other than to the other words.

A co-occurrence matrix such as the one given in Table 2.1 is generally sparse (most values are zero), and high-dimensional because the number of dimensions, which is the number of columns in the matrix, typically corresponds to the vocabulary size of the corpus. There are various ways to tackle such issues, and one way is to apply dimensionality reduction techniques such as Singular Value Decomposition (SVD) on the matrix as a post-processing step. SVD (or any other dimensionality reduction method for that matter) aims to project the word-word co-occurrence matrix into a lower-dimensional approximation matrix, in which the similarity structure between rows (the word representations) are preserved [39, 84].

Moreover, the row co-occurrence frequency counts in the co-occurrence matrix have been found not to be the appropriate measure of association between words [137]. As such, they can be weighted to give more importance to surprising co-occurrences between words and less weight to expected co-occurrences. The idea is that if two vectors share the same surprising event, then it is more discriminative of the similarity between the vectors than the less surprising ones [138]. For example, the context words *dissect* and *exterminate* are more discriminative in measuring the semantic similarity between *mouse* and *rat* than context words such as *have* and *like*. Pointwise Mutual Information (PMI) [31] and its variation Positive Pointwise Mutual Information (PPMI) [115] are two of the more popular ways to formalise the idea of co-occurrences weight. The intuition behind the PMI ($\text{PMI}(x, y) = log\frac{p(x,y)}{p(x)p(y)}$) and the PPMI ($\text{PPMI}(x, y) = max(0, \text{PMI}(x, y))$)

is to measure the strength of the co-occurrence between two words by checking how much more the two words co-occur in a corpus than they would be expected to occur by chance. Therefore, the PMI (or the PPMI) values replace the raw co-occurrence counts in word-word co-occurrence matrices. Further details of the PMI will be discussed later in Chapter 4 (Section 4.3) of this thesis.

On the other hand, KBs alone can be used to obtain word vector representations. In such KB-based approaches, the distributional information of the corpus is not used; hence, they are also alternatively referred to as non-distributional approaches [52]. Non-distributional approaches fall into the traditional paradigm of data representation known as feature engineering, in which human domain experts are required to create representative features for the data. The linguistic properties (e.g. Part-Of-Speech (POS), connotation, etc.) of words and the semantic relations (e.g. synonymy, antonymy, etc.) that exist between words in the KBs are utilised for such hand-crafted word representations. As such, a matrix similar to the word-word co-occurrence matrix given in Table 2.1 can then be built, where the columns (the dimensions) are not the context words but the selected linguistic proprieties and semantic relations that represent the target words that we aim to obtain representation for. A more detailed discussion about the non-distributional word representation approaches is provided later in this chapter (Section 2.3.2) when the KB-based related work is reviewed.

2.3 Related Work

In this section, a comprehensive review of the related work to this thesis is presented. The section commences with a review of corpus-based approaches (Section 2.3.1) where the word representations are learnt solely using text corpora. The corpus-based approaches are categorised into count-based (Section 2.3.1.1), prediction-based (Section 2.3.1.2) and hybrid (Section 2.3.1.3) approaches. That is then followed by Section 2.3.2 in which a discussion on approaches for learning word representations purely from KBs is provided. Next, Section 2.3.3 presents a more focused discussion on the closest line to the work presented in this thesis, namely, the joint learning approaches. The joint approaches in the section are classified, according to the purpose of the joint learning, into additional evidence (Section 2.3.3.1), fine-tuning (Section 2.3.3.2) and sense disambiguation

(Section 2.3.3.3).

2.3.1 Corpus-Based Approaches

Existing corpus-based approaches for learning word representations rely exclusively on the concept of the distributional hypothesis, in which the meaning of words can be represented using the co-occurrences between words in different contexts. Even though there are many different ways to define what contexts are and how to utilise them, the assumptions and the underlying theory are similar. However, corpus-based approaches can be broadly divided into three main categories: (i) count-based, (ii) prediction-based, and (iii) hybrid approaches. In the next three sub-sections, each type of approach is discussed in detail, along with the most notable related work in the literature.

2.3.1.1 Count-Based

Count-based methods have long been known as the traditional way of learning word representations from a corpus. Such methods directly extract the co-occurrence frequency statistics between words in a corpus to represent those words. That is, the word representations are initialised with vectors of co-occurrence *counts*, hence the name. A co-occurrence matrix similar to that of the toy example highlighted earlier in this chapter (given in Table 2.1) forms the basis for all count-based approaches.

Latent Semantic Analysis (LSA) [39] is one of the earliest relevant count-based approach to learn word representations from text corpora. LSA was initially developed for Information Retrieval (IR) systems, in which the interest is more on learning document representations. Therefore, a word-document co-occurrence matrix is created instead of a word-word co-occurrence matrix. SVD is then applied to obtain the final vector representations. However, the principles remain the same as for a word-word co-occurrence matrix because word-document matrices are essentially a special case of word-word matrices [138]. Shortly afterwards came the Hyperspace Analogue to Language (HAL) approach [93]. In HAL, to obtain the representation of a target word, all contexts in which that word appears are analysed and then the co-occurrence count between the target word and each context word is computed and weighted by the distance between the target word and the context words. One drawback associated with the original HAL approach is that the model considers raw co-occurrence counts, without using any measure of association to estimate the strength of the co-occurrence between words. Consequently, high frequent words (e.g. stop words such *the* and *have*, etc.) will have an inappropriately large effect on the obtained vectors. Rohde et al. [125] have identified this problem and tackled it with the Correlated Occurrence Analogue to Lexical Semantics (COALS) model. In COALS, a correlation-based normalisation, which is an association measure, is applied to the co-occurrence matrix to replace the raw co-occurrence count values.

Dhillon et al. [42] also contributed to the count-based approaches with the Low-Rank Multi-View Learning (LR-MVL) model. Word representations in LR-MVL are obtained as a result of the use of Canonical Correlation Analysis (CCA) [69] between the co-occurrence matrices of the left and right context words of a target word. The two matrices are then projected, and the CCA is recursively computed. Lebret and Collobert [87] proposed another count-based method related to the LSA with the difference that the co-occurrence probabilities between words replace the co-occurrence counts. The dimensionality is then reduced using Principle Component Analysis (PCA) [144].

2.3.1.2 Prediction-Based

In recent years, due to the breakthroughs of ML techniques and GPU-based high-performance computing, a large amount of effort has been devoted to prediction-based approaches for learning word representations from text corpora. In prediction-based learning, instead of relying on the co-occurrence counts between words in a corpus, the method learns real-valued, fixed, low-dimensional word vectors such that the learnt vectors can accurately *predict* the co-occurrence between words within a context. This means that the word vectors in such methods are typically randomly initialised and then updated to improve their predictive ability. The principle idea of the prediction-based approaches can be traced back to the development of Neural Network Language Models (NNLMs) when Bengio et al. [17] proposed a large-scale NNLM wherein the task was to predict the next word given the words preceding it in a sentence. The words in their NNLM were represented as real-valued fixed vectors, and a feed-forward neural network was used to compute the joint probability function of the sequence of words in terms of the vectors of those words. Their aim was not to obtain word representations, but to instead improve upon the standard language models by using distributed representations.

Inspired by the NNLMs, prediction-based word representation learning approaches try to predict the co-occurrence between words irrespective of the order of those words in a given context. That is, the preceding and succeeding words of a target word are used to predict the target word. *The learnt word representations by prediction-based methods are often referred to as word embeddings. Thus, henceforth, word embeddings and word representations may be used interchangeably throughout this thesis.*

Skipgram and Continuous Bag-Of-Words (CBOW) [99] models are two very popular prediction-based word representation learning methods that leverage *local* co-occurrences between words in a corpus. Specifically, given a target word, Skipgram tries to find word representations that are able to predict the neighbouring words in a pre-defined co-occurrence context window. More formally, given a training set of text strings $w_1, w_2, w_3, ..., w_{\mathcal{T}}$, and a context window c of a specified size, the objective of the Skipgram model is to maximise the following average log probability:

$$\mathcal{L}_{Skipgram} = \frac{1}{\mathcal{T}} \sum_{i \in \mathcal{T}} \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{i+j}|w_i) \tag{2.1}$$

The conditional probability $p(w_{i+j}|w_i)$ is defined using the softmax function as follows:

$$p(w_{i+j}|w_i) = \frac{\exp(\boldsymbol{w}_{i+j}{}^\top \boldsymbol{w}_i)}{\sum_{w \in \mathcal{V}} \exp(\boldsymbol{w}^\top \boldsymbol{w}_i)} \tag{2.2}$$

Computing this softmax function is computationally costly as it considers all the words in the vocabulary $\mathcal{V}$, and the size of $\mathcal{V}$ is often very large. To overcome this computational problem, Mikolov et al. [101] proposed the Negative Sampling (NEG) technique, which is a variation of Noise-Contrastive Estimation (NCE) [46, 62]. For each target word, NEG trains the model to be able to discriminate between observed (positive) context words and some artificially generated noise (negative) context words. Therefore, the training time is independent of the size

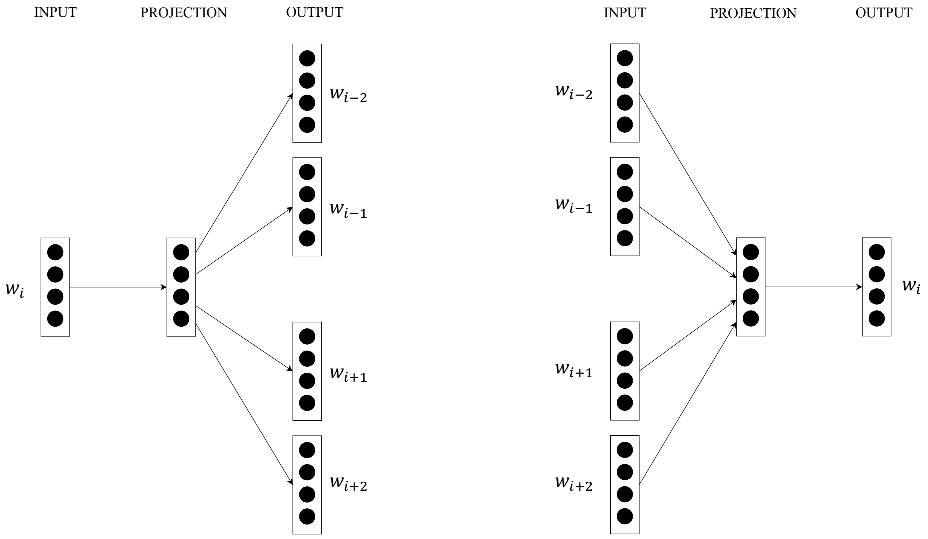


Figure 2.3: Architecture of Skipgram (left) and CBOW (right).

of $\mathcal{V}$. The CBOW model, alternatively, works in a similar manner to the Skipgram model but in the opposite direction. Particularly, unlike in Skipgram, where the context is predicted given the target word, CBOW predicts the target word given the context words in some co-occurrence context window. Hence, the training objective of CBOW is to maximise the following:

$$\mathcal{L}_{CBOW} = \frac{1}{\mathcal{T}} \sum_{i \in \mathcal{T}} \log p(w_i | w_{i-j} ... w_{i+j}) \tag{2.3}$$

Figure 2.3 illustrates the architecture difference between Skipgram and CBOW.

Numerous other prediction-based methods then followed Skipgram and CBOW using them as a basis. For instance, Bojanowski et al. [20] proposed FastText, a method that works on the same basis as Skipgram but learns sub-word (n-gram) embeddings instead of word embeddings. Kenter et al. [78] introduced Siamese-CBOW, which leverages CBOW to learn sentence embeddings rather than word embeddings. Moreover, it is noteworthy that several of the joint approaches for learning word representations that will be thoroughly reviewed later in this chapter (Section 2.3.3) are also based on the Skipgram or CBOW.

The prediction-based approaches are currently the dominant corpus-based approaches for learning word representations in the literature and have been shown to attain a thorough superiority over the count-based approaches [13]. However, prediction-based approaches rely on local co-occurrences between words within a context window. They do not take into consideration the global co-occurrence statistics between words over the entire corpus. Consequently, as noted earlier, pseudo-negative training instances and normalisation over the entire vocabulary are required to compute conditional probabilities. Comparatively, count-based approaches rely on global co-occurrences but suffer from the curse of dimensionality and expensive computational cost. Baroni et al. [13] and Almeida and Xexéo [1] respectively provided a comprehensive systematic comparison and a thorough survey of the count- and prediction-based approaches. Levy and Goldberg [89] showed that there are mathematical links between the two types of approaches. Moreover, to study the relationship between the count- and prediction-based approaches, Bollegala et al. [22] proposed a method for learning a linear transformation between the two.

2.3.1.3 Hybrid Approaches

Taking into account the advantages of using global co-occurrence statistics in count-based approaches and local co-occurrence in prediction-based approaches, Pennington et al. [119] proposed Global Vectors (GloVe), an approach that was a hybrid of the two types. The GloVe works on aggregated global word-word co-occurrence statistics from a corpus. It then predicts the co-occurrence between two words (target and context) using the corresponding word representations. This gives GloVe an advantage in operating on the co-occurrence statistics of the corpus. It is worth mentioning that GloVe will, as we see in the next chapters, form the corpus-based basis for two of the joint approaches proposed in this thesis; thus, a more detailed explanation of GloVe is provided next.

Specifically, GloVe first builds a word-word co-occurrence matrix X whose entries X_{ij} corresponds to the number of times a context word w_j occurs in the context of a target word w_i within the entire corpus. Let $P_{ij} = P(j|i) = X_{ij}/X_i$ be the probability of the context word w_j appears in the context of the target word w_i. To show that the meaning can be directly extracted from the corpus statistics, GloVe uses the ratio of the co-occurrence probability between target and context words, as an alternative to the raw probability between the words. This enables a better distinction between relevant and irrelevant words. In particular, in the concept of thermodynamic phase, for a target word w_i = *ice* and a context word w_j = *steam*, we can examine the relationship of these two words by studying the ratio of their co-occurrence probability with some probe words (e.g. *solid*, *gas*, *water* and *fashion*). For example, for a word w_k = *solid* that is related to *ice* but not to *steam*, we should expect the ratio P_{ik}/P_{jk} to be large. Likewise, with a word w_k = *gas* that is related to *steam* but not to *ice*, the ratio should be small. In the case of w_k words such as *water* and *fashion* that are either related or unrelated to *ice* and *steam*, the ratio should be close to one. Table 2.2 show examples of how the ratio helps to better estimate the relevance between words. We can see that the ratio helps to distinguish the relevant words *solid* and *gas* from the irrelevant words *water* and *fashion.* To reflect the preceding argument to word vector learning, GloVe commences with the following form:

$$F(\boldsymbol{w}_i, \boldsymbol{w}_j, \tilde{\boldsymbol{w}}_k) = \frac{P_{ik}}{P_{jk}} \tag{2.4}$$

To encode the ratio information in the vector space, GloVe opted F to be the vector difference between the target and context words (given that vector spaces are inherently linear structures), and thus Equation 2.4 is modified as follows:

$$F(\boldsymbol{w}_i - \boldsymbol{w}_j, \tilde{\boldsymbol{w}}_k) = \frac{P_{ik}}{P_{jk}} \tag{2.5}$$

Now, the argument of the function F are vectors while the other side is a scalar.

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

Table 2.2: Raw and ratio co-occurrence probabilities of a target word *ice* and a context word *steam* with some probe words [119].

To overcome this issue GloVe takes the dot product of the arguments:

$$F((\boldsymbol{w}_i - \boldsymbol{w}_j)^\top \tilde{\boldsymbol{w}}_k) = \frac{P_{ik}}{P_{jk}} \tag{2.6}$$

The distinction between a target word and a context word in word-word co-occurrence matrix is arbitrary, hence it is possible to exchange the two roles $\boldsymbol{w} \leftrightarrow \tilde{\boldsymbol{w}}$ (and $X \leftrightarrow X^\top$). The model given in Equation 2.6 is not invariant to this relabelling. To solve this issue (and restore the symmetry), GloVe requires F to be homomorphism between the groups $(\mathbb{R}, +)$ and $(\mathbb{R}, \times)$:

$$F((\boldsymbol{w}_i - \boldsymbol{w}_j)^\top \tilde{\boldsymbol{w}}_k) = \frac{F({\boldsymbol{w}_i}^\top \boldsymbol{w}_k)}{F({\boldsymbol{w}_j}^\top \boldsymbol{w}_k)} \tag{2.7}$$

Which is from Equation 2.6 is solved by:

$$F({\boldsymbol{w}_i}^\top \tilde{\boldsymbol{w}}_k) = P_{ik} = \frac{X_{ik}}{X_i} \tag{2.8}$$

And Equation 2.7 is solved by ($F = exp$):

$${\boldsymbol{w}_i}^\top \tilde{\boldsymbol{w}}_k = \log(P_{ik}) = \log(X_{ik}) - \log(X_i) \tag{2.9}$$

In Equation 2.9 the right-hand side would exhibit the symmetry without $\log(X_i)$ which is independent of the word w_k and can be replaced with a bias scalar term b_i for the target word vector $\boldsymbol{w}_i$. Finally, another bias term b_k can be further added for the context word vector $\boldsymbol{w}_k$ to restore the symmetry:

$${\boldsymbol{w}_i}^\top \tilde{\boldsymbol{w}}_k + b_i + \tilde{b}_j = \log(X_{ik}) \tag{2.10}$$

The main drawback associated with the model given by Equation 2.10 is that it treats all co-occurrences between words equally, which can produce noise with very rare or zero occurrences. To overcome this issue, the GloVe final objective is rewritten as a weighted least squares regression model:

$$L_{GloVe} = \sum_{i,j \in \mathcal{V}} f(X_{ij}) \Big({\boldsymbol{w}_i}^\top \tilde{\boldsymbol{w}}_j + b_i + \tilde{b}_j - \log(X_{ij}) \Big)^2 \tag{2.11}$$

Here, f is the weighting function, which will be discussed in further detail later in Section 3.2.1.

Despite the many success stories of prediction-based and hybrid approaches outperforming count-based approaches in numerous NLP tasks such as similarity measurement, named entity recognition, semantic role labelling, and machine translation among many others, these approaches learn word representations purely from large text corpora, ignoring the knowledge encoded in KBs created either manually or semi-automatically over several decades. Therefore, there is no guarantee that the semantic relations existing between words will accurately be captured by the word representations learnt by the corpus-based approaches. In fact, several concerns have been raised in the literature about the ability of such approaches to accurately estimate the strength of the semantic relationships that exist among words [51, 108, 145, 148] or to capture complex linguistic phenomena such as task- or domain-specificity [7, 23, 29, 111] and ambiguity [73, 76, 77, 126]. Section 1.2 of the previous chapter provided concrete examples of those concerns.

2.3.2 KB-Based Approaches

Manually created KBs such as WordNet [104] and FrameNet [10] provide valuable information about words meanings, by defining a word using its linguistic properties and the semantic relations it has with other words. As such, KBs can be solely used to obtain the vector representations of words. KB-based approaches are regarded as non-distributional approaches, because there is no corpus involved in constructing the word representations (i.e. no distributional information). KB-based approaches are examples of the traditional method of data representation, in which we need human domain experts to produce representative features of the data. Consequently, the research on using KBs alone for learning word representations in vector space has been limited. To the best of our knowledge, there are only two methods that exclusively aim to obtain the vector representation of words purely from KBs.

Faruqui and Dyer [52] proposed a method that constructs word vector representations purely from KBs. Specifically, the method obtains word representations by extracting linguistic features from KBs. Such features include the word's *POS, SuperSense (SS), Sentiment Polarity (POL), Colour Association (COL)* and

Word	POL.POS	COLOUR.PINK	SS.NOUN.FEELING	PTB.VERB	...	ANTO.FAIR
love	1	1	1	1		0
hate	0	0	1	1		0
ugly	0	0	0	0		1
beauty	1	1	0	0		0
refundable	0	0	0	0		0

Table 2.3: Examples of the non-distributional word vectors [52].

Antonym (ANT), to name a few. Next, each dimension in the vector representation corresponds to a linguistic feature having a value 0 or 1, demonstrating the presence or absence of that feature in the word. For this feature extraction, they used several KBs to extract 172, 418 linguistic features (dimensions). Table 2.3 shows examples of some non-distributional word vectors. SVD is then performed on the linguistic matrix to obtain lower-dimensional word vectors.

Goikoetxea et al. [60] introduced a RandomWalk approach for learning word representations solely from a KB. The method aims to utilise structural information in the KB to encode the word meaning. That is, to use random walks in a KB as a way of encoding the meaning instead of the distributional information in a corpus. Specifically, given a KB specifying semantic relations (edges), the method first randomly chooses a word in the KB and performs a random walk starting from the selected word. At each random walk, the algorithm might, based on an assigned probability, stop at a particular word or continue to a random neighbour word. This being so, each random walk can be seen as a context for the words in the vocabulary. The words that were visited during the random walk are sequentially recorded to generate a pseudo corpus. Examples of sentences in the generated pseudo corpus are: In the first sentence, the random walk starts with the word *amphora* followed by what it is usually filled with, *wine*, and then a certain bottle size, *nebuchadnezzar* and ending with words associated with wine storage, such as *bear*, *retain* and *long*. The random walk proceeds similarly with the second sentence. Those two sentence examples gave an indication of the im-

- "*amphora wine nebuchadnezzar bear retain long*".
- "*graphology writer write scribble scrawler heedlessly in haste jot note*".

plicit semantic information that can be found in the generated pseudo corpus. After this pseudo corpus is constructed from the entire KB, it is then fed to a corpus-based model (Skipgram and CBOW were used in this particular model) to learn the word representations.

As highlighted earlier, although KBs provide invaluable information about the meaning of words, such information is manually curated and thus costly to produce. Similarly, exploiting such information to obtain the vector representations of words would as well require manual effort for feature engineering. Additionally, in a KB, a particular word often has a limited number of entries, which makes it difficult to estimate the strength of the relation between two words when learning their representations. Moreover, new words or novel uses of existing words are not very well covered by the manually constructed and maintained KBs.

2.3.3 Joint Approaches

From the above (and the detailed discussion in Section 1.2), it is apparent that using only a single type of resource, either text corpora or KBs, for learning word representations comes with several limitations. Consequently, there has been an exploration of joint approaches that learn word representation from both resources. *The work presented in this thesis is closely related to this kind of methods.* In what follows in this section, we thoroughly review joint approaches closely related to the three joint approaches proposed in this thesis. In particular, they are categorised according to the purpose of the joint learning. That is, they are categorised as joint approaches for: (i) additional evidence, (ii) fine-tuning, and (iii) sense representations. It should be mentioned that insight into each category (with concrete examples) has been provided in detail in Section 1.2.

2.3.3.1 Additional Evidence

As already noted, word representations learnt by corpus-based approaches are exclusively dependent on the co-occurrence statistics between words. However, such a statistical nature might be insufficient when some words rarely occur in the corpus. This means that, for example, two synonym words may lack enough context for the strength of the relationship between them to be estimated accurately, which may result in inadequate representations being learned. However, a KB that explicitly defines the semantic relations between words can provide *additional evidence* for corpus co-occurrence and thus pull similar words closer together and learn similar representations for them.

For additional evidence, Yu and Dredze [148] proposed a Relation Constrained Model (RCM) that uses a KB to improve word representations learnt via a corpus prediction-based approach. The RCM uses word similarity information in a KB to improve the word representations learnt using CBOW. It assigns high probabilities to words that are listed as similar in the KB. Specifically, given a KB that indicates the relations between words, RCM defines an objective to learn the word representations that predict a word from another related word. Next, RCM was linearly combined with the CBOW objective to form a joint model that provides further evidence that two words co-occurring within context-window in the corpus with an indication of how well the two words are related in the KB. Although RCM jointly learns the word representations using a corpus and a KB, it still suffers from some limitations. For example, only synonym relations are considered in RCM, whereas, as we see in the next chapter, other semantic relations are helpful for improving the learnt word representations. Moreover, in RCM, the CBOW objective is used, which is a prediction-based approach that considers only the local co-occurrence between words without utilising the global co-occurrences over the entire corpus.

Similarly, Xu et al. [145] proposed the RC-NET framework, which leverages knowledge existing in KBs to enforce knowledge constraints during word representation learning by Skipgram. Specifically, RC-NET classifies the knowledge extracted from the KB into Relational (R-NET) and Categorical (C-NET) knowledge. Relations such as *is-a*, *part-of* and *child-of* are regarded as relational knowledge, whereas relations like *gender*, *location* and *synonyms* are classified as categorical knowledge. RC-NET is then accordingly constructed of three models based on the knowledge category. For the R-NET objective, the principle is to represent words and relations in the same embedded space. Specifically, given a relational triplet (w_i, r, w_j) where a semantic relation r exists in the KB between two words w_i and w_j, the aim is to learn the vector representations $\boldsymbol{w}_i$, $\boldsymbol{w}_j$, and $\boldsymbol{r}$ such that $\boldsymbol{w}_j$ is close to $\boldsymbol{w}_i + \boldsymbol{r}$ in the space if the relation r holds between the two words w_i and w_j, otherwise, $\boldsymbol{w}_j$ would be pushed away from $\boldsymbol{w}_i + \boldsymbol{r}$. For the C-NET objective, the idea is to enforce the constraint that if the two words w_i and w_j

belong to the same category (e.g. synonyms) in the KB, then they should be close to each other in the space. The R-NET and C-NET objectives are then combined with the Skipgram objective to formulate the joint RC-NET framework. Similar to RCM, RC-NET is limited to using local co-occurrence counts.

Furthermore, Faruqui et al. [51] introduced a retrofitting (Retro) model that uses a corpus and a KB for learning word representations in a post-processing manner. Retro works on refining pre-trained word representations learnt via a corpus-based method with relational information extracted from a KB. Methods analogous to Retro (i.e. utilising KBs in a post-processing manner) are often referred to as *retrofitting* line of work in the literature. In Retro, W is a given matrix consisting of word representations $(\boldsymbol{w}_1, ..., \boldsymbol{w}_n)$ learned using a corpus-based approach, and a KB encodes the related words. The Retro objective is to learn a matrix $\hat{W} = (\hat{\boldsymbol{w}}_1, ..., \hat{\boldsymbol{w}}_n)$ such that the columns in $\hat{W}$ are close to their counterparts in W and their related words in the KB. Particularly, to learn a word representation $\hat{\boldsymbol{w}}_i$ so that it is close to the observed (pre-trained) $\boldsymbol{w}_i$ and its related word $\hat{\boldsymbol{w}}_j$.

Similar to Retro, Mrkšić et al. [108] presented ATTRACT-REPEL, which is a retrofitting model that uses both a KB and a corpus to learn word representations. It uses a set of synonyms and antonyms extracted from a KB to derive constraints that refine the word representations learnt via a corpus-based approach. Specifically, given pre-trained word vectors representations, ATTRACT-REPEL refines the vector space by forcing the vectors of synonyms to be located close to each other in the vector space (ATTRACT), and separating the vectors of antonyms (REPEL).

The retrofitting line of work is attractive because it can be used to fit arbitrary pre-trained word representations to an arbitrary KB, without having to retrain the word representations. However, a disadvantage of such an approach is that we cannot use the rich information in the KB during the learning phase of the word representations from the corpus. Moreover, incompatibilities between the corpus and the KB, such as missing terms, must be carefully considered.

2.3.3.2 Fine-Tuned Word Representations

One of the strengths of combining text corpora and KBs becomes evident when the learnt word representations can be fine-tuned to encode a particular semantic relation for a specific task. For example, encoding the hierarchical information that exists between words in the learnt word representations has been shown to be vital for various NLP tasks such as textual entailment [35], text generation [19] and question answering [72], to name but a few. Corpus-based word representations have shown some capacity for encoding hierarchical information. However, those word representations were not explicitly trained for that purpose, and they have been found to be inadequate for several tasks, as we see later in Chapter 5. As a result, the emergence of joint approaches centring on learning fine-tuned word representations for hierarchical information has been witnessed in the literature [111, 114, 140, 149].

For example, Nguyen et al. [111] proposed a neural model called HyperVec that learns hierarchical word representations utilising both a corpus and a KB. To encode the hierarchical information in the learnt word representations, HyperVec learns to move hypernym and hyponym vectors close to each other in the vector space to strengthen their distributional similarity in comparison with other relations, and generates a distributional hierarchy between them. The assumption HyperVec was built upon, to encode a distributional hierarchy, is that a context word that appears with both the hypernym and hyponym words gives an indication of which is semantically more general. For example, assume that we have a hypernym pair (w, u), where the hyponym word w is *bird*, the hypernym u is *animal*, and we have a common context word c, which is *flap*. The context word *flap* can be seen as a distinctive characteristic of *bird* but not of *animal*, because not all animals can flap. Therefore, the model defines an objective to enforce the similarity between *bird* and *animal* as well as to decrease the distributional generality of *animal* by moving *bird* closer to it. In the case where the context word c is a distinctive characteristic of a hypernym v but not of a hyponym w, then the aim is to decrease the distributional generality of w. For example, given the hypernymy pair *(bird, animal)*, the context word *rights* is a more distinctive characteristic of *animal* than of *bird*, hence it should be closer to *animal*. Another objective is then defined to enforce this. In the final step, the two objectives are combined with Skipgram to form the final HyperVec model objective.

Anh et al. [7], likewise, introduced a Dynamic Weighting Neural Network model (DWNN) to learn fine-tuned word representations for hierarchical information using a KB and a corpus. The approach commences by extracting a set of hypernym pairs from a KB, and then using the extracted pairs to collect contextual words that appear with those pairs in the same sentence in the corpus to form the training triples. Let (w_i, w_j) be the hypernymy pair where w_i is the hyponym and w_j is the hypernym and $c_1, c_2, ..., c_k$ are the context words that appear with the hypernymy pair in the same sentence. The training triples are then in the form $(w_i, w_j, (c_{1ij}, c_{2ij}, ..., c_{kij}))$. The DWNN is next explicitly designed to learn word representations that are able to predict hypernym words from the hyponyms and the contextual words. Moreover, Nickel and Kiela [114] proposed the Poincaré Ball model for learning hierarchical embeddings into hyperbolic space. The Poincaré Ball model makes use of the WordNet hypernymy methods and learns embeddings based on n-dimensional Poincaré Balls from a taxonomy, without any information from the corpus.

On the other hand, Vulić and Mrkšić [140] presented a retrofitting approach called Lexical Entailment Attract-Repel (LEAR). It is an extended version of the post-processing model ATTRACT-REPEL, and is used to encode the asymmetric relation (hypernymy) of lexical entailment between words jointly with the symmetric relations. Specifically, given a set of hypernymy pairs (w_i, w_j) extracted from a KB, LEAR defines an objective to rearrange the vector norms of the words to encode hierarchical information, i.e. the more general concepts/words (hypernyms) are assigned larger norms than the narrower concepts (hyponyms). This objective is then combined with the ATTRACT-REPEL to construct the final LEAR model.

A common drawback associated with the aforementioned methods is that they mainly focus on *pairwise* hypernymy relations, ignoring the *full* hierarchical path. The full hierarchical path of hypernymy, as we see later in Chapter 5, not only gives a better understanding of the hierarchy than a single hypernymy edge but also has been empirically shown to be useful for various tasks.

2.3.3.3 Sense Representations

One of the most common drawbacks associated with existing methods for learning word representations is that they are built to learn a single vector representation per word, neglecting the possible multiple senses/meanings of a word [71]. For example, consider the ambiguous word *bank* that could mean either a *financial institution* or a *river bank*. The two senses of *bank* are essentially different, and embedding both senses to the same point is inadequate. As a result, different solutions have been proposed in the literature to tackle this problem by learning *sense representations*, which examine the sense-related information of words.

For example, Reisinger and Mooney [124] proposed a method for learning sense-specific high-dimensional distributional vector representations of words, which was later extended by Huang et al. [71] using global and local contexts to learn multiple sense embeddings for an ambiguous word. Similarly, Neelakantan et al. [110] presented the Multi-Sense Skipgram (MSSG), an online cluster-based sense representation learning method, by extending Skipgram. Unlike Skipgram, which updates the gradient of the word vector according to the context, MSSG predicts the nearest sense first and then updates the gradient of the sense vector. MSSG was then enhanced with a Non-Parametric version (NP-MSSG) [110] that estimates the number of senses per word and learns the corresponding sense representations instead of learning a fixed number of senses per word. Both MSSG and NP-MSSG rely on the corpus to learn sense representations.

Furthermore, inspired by the Retro approach, Jauhar et al. [76] proposed a similar post-processing sense retrofitting (SenseRetro) approach that takes arbitrary pre-trained word vectors learnt using a corpus-based approach and retrofits them to generate sense vectors leveraging sense knowledge in a given KB. Particularly, let W be a matrix that contains a collection of pre-trained word vector representations $(\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_n)$, and let a KB connect a word w_i to its senses s_{ij} by an edge $w_i - s_{ij}$ and connect senses that have a semantic relation with other senses by an edge $s_{ij} - s_{i'j'}$. The SenseRetro objective is then to learn the matrix $S = (\boldsymbol{s}_{11}, \boldsymbol{s}_{12}, ..., \boldsymbol{s}_{nm})$ for word senses that are consistent with the observed vectors W and the KB connections.

Iacobacci et al. [73] proposed a sense-specific (SenseEmbed) word representations approach leveraging a KB alongside a text corpus. SenseEmbed works upon the principle that if we have a large sense-annotated corpus, then a typical word representation approach can be used to learn sense-aware word representations. To this end, SenseEmbed considers a large sense-unannotated corpus and runs a Word Sense Disambiguation (WSD) system to sense-annotate the corpus automatically. Next, CBOW is used to learn the sense and word representations from the automatically annotated corpus. As a final step, a KB is utilised to associate the learnt words with their related senses. Similarly, Camacho-Collados et al. [28] used the knowledge in WordNet and Wikipedia. They use the contextual information of a particular concept from Wikipedia and WordNet synsets prior to learning two separate vector representations for each concept. Likewise, Jauhar et al. [76] present Skipgram-WSD, an approach quite similar to SenseEmbed, but instead of using the CBOW objective to learn the word representations, they opted to use Skipgram on the automatically sense-annotated corpus.

The above-mentioned approaches use either a corpus, a corpus with a KB (sense inventories defining the different senses of a word), or word-sense taggers that can be applied on unlabelled corpora to generate automatic sense-labelled training data. None of the prior work on learning sense-aware word representations has attempted to use manually sense-labelled corpora (KB's linked senses) jointly with the unlabelled corpora. As such, it remains unclear whether unlabelled data can help the process of learning sense representations, which indeed is the case (unlabelled data can help) as we see later in Chapter 6.

2.4 Summary

This chapter has provided the necessary background to the essential research areas and the related work of this thesis. Representation learning and word representation learning areas were presented as the foundation of the work presented in this thesis. From that, numerous related word representation learning approaches were presented and comprehensively reviewed.

The chapter began with a discussion of the principles of data representation and how it plays a significant role in any information processing tasks, and in ML and NLP tasks specifically. The difference between traditional data representations and representation learning theory was then highlighted, which paved the way for introducing word representation learning. The distributional hypothesis and VSM were next discussed as the cornerstones of word representation learning from text corpora. That was followed by discussing the way KBs can also be leveraged to obtain word representations.

The chapter then presented a thorough review of the work related to this thesis. It commenced with a review of the corpus-based approaches where the corpus is solely used to learn the word representations. The corpus-based approaches were classified into count-based, where the words are represented directly from the co-occurrence statistics, prediction-based, in which the word representations are learnt to predict the co-occurrences, and hybrid count- and prediction-based approaches. Next, a discussion of word representations obtained solely from KBs (KB-based approaches) was then presented. The chapter was then concluded with a more focused discussion on the joint learning approaches (joint approaches for additional evidence from the KBs, fine-tuned word representation and sense representations), which is the closest line to the work presented in this thesis.

The following chapter introduces the first of a number of joint approaches that use a corpus and a KB for learning word representations presented in this thesis. Specifically, it introduces a joint word representation learning that uses additional evidence from KBs.

Chapter 3

Joint Representation Learning for Additional Evidence

3.1 Introduction

In the previous chapter, a number of methods that have been proposed to learn word representation were presented. Among which we discussed methods for learning word representations using the information distributed in a text corpus alone (Section 2.3.1), and how they have proved to be valuable in various NLP tasks. However, it was also emphasised that despite the many success stories of those corpus-based methods in learning word representations and capturing the word meaning, they operate at surface-level word co-occurrences, and therefore disregard the rich semantic relational information between two words that might be encoded in KBs. Ignoring such kind of relational information that exists in the KBs between words that co-occur together in the corpus can be problematic in different scenarios [51, 92, 148]. For example, the corpus might not be sufficiently large to obtain reliable word co-occurrence counts, which is problematic when learning representations for rare words [145]. Moreover, some dissimilar words may share similar contexts, such as synonyms and antonyms. However, a KB can easily discriminate between such semantic relations [92].

Consequently, a line of methods that go beyond using only the corpus to learn word representations and combine it with KBs has gained much popularity lately. Such joint methods, as discussed earlier (Section 2.3.3), are generally classified into two classes, joint and retrofitting. Methods on both classes are often built upon the corpus-based approaches, but use semantic relations available in KBs, as an *additional evidence* alongside the information provided by the distributional co-occurrence. They seek to enhance the learnt word representations by pulling the representations of similar and related words closer together.

However, several limitations are associated with prior work (discussed in details in Section 2.3.3.1) that use KBs with the corpus for *additional evidence*. For example, in RCM [148] only the synonym relation was considered from the KB, whereas different types of semantic relations might be useful. In addition, RCM uses only a prediction-based objective, CBOW, as its basis, which considers only local co-occurrences, which is also the case in RC-NET [145]. Although retrofitting approaches such as Retro [51] and Attract-Repel [108] offer an attractive option because they can be used to fit arbitrary pre-trained word representations to an arbitrary KB, a disadvantage of such approaches is that we cannot use the rich information in the KB during the learning phase of the word representations from the corpus. Incompatibilities between the corpus and the KB need to be carefully considered in such retrofitting approaches too.

In this chapter, we propose a new method called *Joint Representation Learning for Additional Evidence* (JointReps), the first of a number of *joint* approaches for learning word representations using a corpus and a KB presented in this thesis. In the proposed JointReps, we aim to address the limitations of prior work discussed above. Firstly, instead of using only synonyms, JointReps uses different types of semantic relations that exist in the KB. As we show later (Section 3.4), besides synonyms, numerous other semantic relations are useful for different tasks. Secondly, rather than using a purely prediction-based objective that considers only local co-occurrences, in JointReps we use a hybrid of count- and prediction-based objective that utilises the global co-occurrences over the entire corpus. Indeed, it has been shown that one can learn superior word representations by using global co-occurrences instead of local co-occurrences [119]. Moreover, unlike the *retrofitting* approaches, JointReps *jointly* learns from the corpus and the KB, allowing it to benefit from the KB during the learning process.

The remainder of this chapter is organised as follows. Section 3.2 provides details of the learning process of JointReps. This is followed by presenting the experimental setup used to train JointReps (Section 3.3). Then, in Section 3.4, an extensive evaluation of the method is presented. Finally, Section 3.5 provides a summary of the material presented in this chapter.

3.2 Learning Process of JointReps

We propose JointReps, a method to learn word representations from both a corpus and a KB in a *joint* manner. First, in Section 3.2.1, we briefly review GloVe [119], which forms the basis of the corpus-based objective in JointReps. Next, Section 3.2.2 describes how we incorporate the KB by deriving some linguistic constraints. Finally, in Section 3.2.2, we detail the *joint* learning method.

3.2.1 Corpus-Based JointReps

We use GloVe (discussed in further details earlier in Section 2.3.1.3 of this thesis) as the corpus-based method of the proposed JointReps. GloVe learns word vector representations from a text corpus by leveraging statistical information computed from a global word co-occurrence matrix. In particular, given a corpus $\mathcal{C}$, GloVe commences by creating a co-occurrence matrix $\mathbf{X}$, where each *target word* (i.e. the word that we want to learn a representations for) is represented by a row in $\mathbf{X}$, and the *context words* that co-occur with it in some contextual window, are represented by the columns of $\mathbf{X}$. The entries X_{ij} denote the total occurrences of target word w_i and the context of word $\tilde{w}_j$ in the corpus. Next, for each word w_i in the vocabulary $\mathcal{V}$ (i.e. the set of all words in the corpus), GloVe seeks to learn word representations $\boldsymbol{w}_i, \tilde{\boldsymbol{w}}_i \in \mathbb{R}^d$ corresponding respectively to whether w_i is a target word or a context word $\tilde{w}_i$. Here, the boldface $\boldsymbol{w}_i$ denotes the word representation (vector) of the word w_i, and the dimensionality d is a user-specified hyperparameter. The GloVe representation learning method minimises the following weighted least squares loss:

$$J_{\mathcal{C}} = \frac{1}{2} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} f(X_{ij}) \Big({\boldsymbol{w}_i}^\top \tilde{\boldsymbol{w}}_j + b_i + \tilde{b}_j - \log(X_{ij}) \Big)^2 \tag{3.1}$$

Here, the two real-valued scalars b_i and $\tilde{b}_j$ are biases associated respectively with w_i and $\tilde{w}_j$. The weighting function f assigns a lower weight for extremely frequent co-occurrences to prevent over-emphasising such co-occurrences, and is given by:

$$f(t) = \begin{cases} (t/t_{\max})^{\alpha} & \text{if } t < t_{\max} \\ 1 & \text{otherwise} \end{cases} \tag{3.2}$$

The GloVe objective function defined in Equation 3.1 attempts to predict the co-occurrences between two words w_i and $\tilde{w}_j$ using the inner-product between the corresponding embeddings $\boldsymbol{w}_i$ and $\tilde{\boldsymbol{w}}_j$. Those embeddings are learnt such that the squared difference between the inner-product and the logarithm of their co-occurrence count is minimised. That is, Equation 3.1 is designed such that the learnt words embeddings represent the relationship between two words by their vector difference [141].

3.2.2 Incorporating the KB

We would like the proposed JointReps to learn the word representations from both the corpus and the KB. However, GloVe is a corpus-only method that does not leverage any available KBs. Therefore, it is likely to encounter problems when learning word representations from rare co-occurrences and may fail to capture the desired semantics. To address this problem, we combine the KB in the learning process by deriving constraints from the KB that must be satisfied by the learnt word representations. Specifically, given a KB $\mathcal{S}$, we define an objective $J_{\mathcal{S}}$ that considers not only two-way co-occurrences between a target word w_i and one of its context words $\tilde{w}_j$, but rather a three-way co-occurrence between w_i, $\tilde{w}_j$ and the semantic relations R that exists between them in the KB. Thus, the KB-based objective is defined as follows:

$$J_{\mathcal{S}} = \frac{1}{2} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} R(w_i, w_j) \left(\boldsymbol{w}_i - \tilde{\boldsymbol{w}}_j\right)^2 \tag{3.3}$$

Here, $R(w_i, \tilde{w}_j)$ is a binary function that returns 1 if the semantic relation R exists between the words w_i and $\tilde{w}_j$ in the KB, 0 otherwise. The objective given by Equation 3.3 enforces the constraint that the words that are connected by a semantic relation R in the KB must have similar word representations.

3.2.3 Joint Objective

To formalise the final objective function of JointReps, we would like to learn target and context word representations $\boldsymbol{w}_i$, $\tilde{\boldsymbol{w}}_j$ that simultaneously minimise both the corpus-based objective function (Equations 3.1) and the KB-based objective (Equation 3.3). Therefore, we defined a combined objective as their linearly weighted combination given by:

$$J_{JR} = J_{\mathcal{C}} + \lambda J_{\mathcal{S}}. \tag{3.4}$$

Here, $\lambda \in \mathbb{R}^+$ is a regularisation coefficient that determines the influence imparted by the KB on the word representations learnt from the corpus. Details of estimating the optimal value of λ is described later in Section 3.3.2.

The overall objective function given by Equation 3.4 is non-convex with respect to the four variables $\boldsymbol{w}_i$, $\tilde{\boldsymbol{w}}_j$, b_i and $\tilde{b}_j$. However, if we fix three of those variables, then the objective function becomes convex in the remaining one variable. We use an alternative optimisation approach where we first randomly initialise all the parameters and then cycle through the set of variables in a pre-determined order updating one variable at a time while keeping the other variables fixed.

The derivatives of the objective function with respect to the four variables are given as follows:

$$\begin{aligned}
\frac{\partial J_{JR}}{\partial \boldsymbol{w}_i} = & \sum_j f(X_{ij}) \tilde{\boldsymbol{w}}_j \left({\boldsymbol{w}_i}^\top \tilde{\boldsymbol{w}}_j + b_i + \tilde{b}_j - \log(X_{ij}) \right) \\
& + \lambda \sum_j R(w_i, w_j)(\boldsymbol{w}_i - \tilde{\boldsymbol{w}}_j) && (3.5) \\
\frac{\partial J_{JR}}{\partial b_i} = & \sum_j f(X_{ij}) \left({\boldsymbol{w}_i}^\top \tilde{\boldsymbol{w}}_j + b_i + \tilde{b}_j - \log(X_{ij}) \right) && (3.6) \\
\frac{\partial J_{JR}}{\partial \tilde{\boldsymbol{w}}_j} = & \sum_i f(X_{ij}) \boldsymbol{w}_i \left({\boldsymbol{w}_i}^\top \tilde{\boldsymbol{w}}_j + b_i + \tilde{b}_j - \log(X_{ij}) \right) \\
& - \lambda \sum_j R(w_i, w_j)(\boldsymbol{w}_i - \tilde{\boldsymbol{w}}_j) && (3.7) \\
\frac{\partial J_{JR}}{\partial \tilde{b}_j} = & \sum_i f(X_{ij}) \left({\boldsymbol{w}_i}^\top \tilde{\boldsymbol{w}}_j + b_i + \tilde{b}_j - \log(X_{ij}) \right) && (3.8)
\end{aligned}$$

3.3 Experiments

The experimental settings that have been used to train the proposed JointReps is given in this section. The section is organised as follows. It commences, Section 3.3.1, with detailed information about the corpus and the KB used for training the JointReps along with the applied data preparation. Section 3.3.2 then explains the model setup and the training details with respect to the model's hyperparameters and the optimisation process.

3.3.1 Training Data

We used ukWaC [54] as the corpus in our experiments. It is a large English Web corpus comprising of approximately 2 billion tokens crawled from the Web from *.uk* internet domain. UkWaC was collected to be a resource of traditional general language containing wide range of text types and topics. Thus, comprising pre-web texts of different nature that are available in electronic format on the Web (including the likes of sermons, recipes, technical manuals, short stories and transcript of spoken language, among others), as well as web-based texts such as blogs, personal pages or postings in forums.

We used the WordNet [104] as the KB in our experiments. From the WordNet, we consider *seven* different relation types (synonym, hypernym, hyponym, member-holonym, member-meronym, part-holonym and part-meronym). For synonymy, we generate all the pairwise combinations of words in a given synset to create synonymous word-pairs. For other relations, we consider two words u and

Relation	**Edges**	**Examples**
Synonym	87,060	(*scream*, *screech*), (*greatness*, *immensity*)
Hypernym	119,029	(*ostrich*, *bird*), (*flower*, *plant*)
Hyponym	122,926	(*car*, *coupe*), (*book*, *storybook*)
Member-holonym	11,506	(*portugal*, *europe*), (*policeman*, *police*)
Member-meronym	11,431	(*company*, *crew*), (*france*, *frenchman*)
Part-holonym	13,082	(*mouth*, *face*), (*minute*, *hour*)
Part-meronym	13,251	(*door*, *lock*), (*theater*, *stage*)

Table 3.1: Number of edges and some selected examples for different relation types extracted from the KB (WordNet).

v connected by a semantic relation R if R exists between the two synsets encompassing u and v. See Table 3.1 for detailed information regarding the extracted relations from WordNet.

It is worth noting that although we use WordNet as a concrete example of a KB in this work, there are no assumptions made regarding any structural properties unique to a particular KB. In fact, any KB that defines pairwise semantic relations between words such as FrameNet [10] and the Paraphrase Database (PPDB) [58] can be used with the proposed method.

3.3.2 Model Setup and Training

Building the co-occurrence matrix $\mathbf{X}$ is essential step for the proposed JointReps. We first create the word co-occurrence matrix $\mathbf{X}$ considering the words that occur at least 20 times in the corpus to reduce any potential noise such as misspelling words. Following prior recommendations [89], we set the context window to the 10 tokens preceding and succeeding a target word in a sentence and extract unigrams as context words. Co-occurrences are weighted by the inverse of the distance between the target word and a context word, measured by the number of tokens appearing in between. We adopt a harmonic weighting function using the reciprocal $\frac{1}{d}$ of the distance between two co-occurrences. For example, a context word co-occurring 5 tokens from a target word would contribute to a co-occurrence count of $\frac{1}{5}$. The weighting function given by Equation 3.2 is computed with $\alpha = 0.75$ and $t_{\max} = 100$. We use Stochastic Gradient Descent (SGD) as the optimisation method.

The overall algorithm of the JointReps is listed in Algorithm 1. The word representations are randomly initialised to the uniform distribution in the range $[-1, +1]$ for each dimension separately. Experimentally, $T = 20$ iterations were found to be sufficient for the proposed method to converge to a solution.

Algorithm 1 in Line 3 iterates over the nonzero elements in $\mathbf{X}$. The estimated overall time complexity for n nonzero elements is $\mathcal{O}(|\mathcal{V}|dTn)$, where $|\mathcal{V}|$ denotes the number of words in the vocabulary. Typically, the global co-occurrence matrix is highly sparse, containing less than 0.03% of non-zero entries. It takes around 50 minutes to learn 300 dimensional word representations for $|\mathcal{V}|= 434,826$ words ($n = 58,494,880$) from the ukWaC corpus on a Xeon 2.9GHz 32 core 512GB RAM

Algorithm 1 JointReps learning.

Input: Word co-occurrence matrix $\mathbf{X}$ specifying the co-occurrences between words in the corpus $\mathcal{C}$, relation function $R(w_i, \tilde{w}_j)$ specifying the semantic relations between words in the KB $\mathcal{S}$, dimensionality d of the word representations, and the maximum number of iterations T.

Output: Vector Representations $\boldsymbol{w}_i, \tilde{\boldsymbol{w}}_j \in \mathbb{R}^d$, of all words $w_i, w_j \in \mathcal{V}$.

```
1:  Initialise word vectors $\boldsymbol{w}_i, \tilde{\boldsymbol{w}}_j \in \mathbb{R}^d$ randomly.
2:  for $t = 1$ to $T$ do
3:      for $(i, j) \in \mathbf{X}$ do
4:          Use Equation 3.5 to update $\boldsymbol{w}_i$
5:          Use Equation 3.6 to update $b_i$
6:          Use Equation 3.7 to update $\tilde{\boldsymbol{w}}_j$
7:          Use Equation 3.8 to update $\tilde{b}_j$
8:      end for
9:  end for
10: return $\boldsymbol{w}_i, \tilde{\boldsymbol{w}}_j \quad \forall w_i, w_j \in \mathcal{V}$.
```

machine. For each word w_i, JointReps learns a target representation $\boldsymbol{w}_i$, and a context representation $\tilde{\boldsymbol{w}}_i$. Prior work in learning word representations [89] shows that the addition of the two vectors $\boldsymbol{w}_i + \tilde{\boldsymbol{w}}_i$, gives a better representation for the word w_i. In our experiments, we followed this prior recommendation and created the final representation for a word by adding its target and context representations. In the remaining sections of this chapter, we consider those word representations.

As a concrete example of how JointReps works, let us assume that the target word w_i and the context word w_j are *bird* and *vertebrate* respectively. Line 3 in Algorithm 1 first verifies their co-occurrence in the corpus. Then Lines 4 and 6 look up the semantic relation that exists (if any) between *bird* and *vertebrate* in the KB before updating the corresponding word vectors and the bias terms (Lines 5 and 7). If the two words have a semantic relation R, which is the case in this example, JointReps will enforce the constraint that their learnt word representations must be similar. As such, the KB provides additional evidence to the co-occurrence statistics obtained from the corpus to embed *bird* and *carnivore* close to each other in the vector space. In the case where the two words, target and context, are not recorded with any semantic relation R in the KB, the corresponding vector representations will be learnt solely from the corpus. That is, the function $R(w_i, \tilde{w}_j)$ in Equation 3.3 will returns 0 and the word representations will be learnt purely from GloVe.

3.4 Evaluation and Results

We evaluate the proposed JointReps on two standard tasks: Word Similarity measurement (Section 3.4.1) and Word Analogy prediction (Section 3.4.2). In each task, we study the effectiveness of using the corpus and the KB jointly for learning word representations covering a wide range of relation types by comparing the performance of JointReps against a corpus-only baseline. Next, in Section 3.4.3 we report a comparison between the JointReps and previously proposed methods that learn word representations from both a corpus and a KB. In Section 3.4.4, we evaluate the effect of the corpus size on the performance of the JointReps. Finally, Section 3.4.5 provides an evaluation of the effect of the dimensionality on the word representations learnt by the JointReps.

The best performance for each task in each of the upcoming results tables is shown in **bold** and statistical significance is indicated by asterisk (*). Moreover, it is worth reminding that the all performance results of the proposed JointReps reported in the forthcoming tables are obtained by training the model following the experimental setup discussed above in Section 3.3.

3.4.1 Word Similarity Measurement

The word similarity measurement task is one of the most popular methods to evaluate the word vector representations. Before discussing the evaluation results, the section commences with a brief description of the task and how it is used to evaluate the word representations (Section 3.4.1.1). Section 3.4.1.2 then provides details of the benchmark datasets that have been used in this task. The results are then provided with a detailed discussion in Section 3.4.1.3.

3.4.1.1 Task Description

The inception of word similarity measurement task dates back to 1965 [127] when the human judgements on word semantic similarity firstly involved in an experiment to test the distributional hypothesis [9]. The task is based on the idea that we can evaluate the distances between words in the vector space through human judgements. Hence, numerous benchmark datasets are available with human judgements (annotations) about the semantic similarity between words. Specifically, the human annotators are given a set of word-pairs and are asked to rate the degree (on a scale of $0 - 10$ or any other scale depending on each individual dataset) of similarity of each pair. For example, the pair (*street*, *avenue*) receives 8.87 average similarity rating, whereas the pair (*sugar*, *approach*) receives only 0.88. Next, the cosine similarity between word vector representations learnt by a particular method for two words in a benchmark dataset is measured and then compared against the average similarity ratings given by the human annotators for those two words. If there is a high degree of correlation between human similarity ratings and the similarity scores computed using the learnt word representations, we can conclude that the word representations capture word semantics as perceived by humans.

Dataset	**Size**	**Scale**	**Examples**
RG	65	0-4	(*automobile*, *car*, 3.92)
			(*car*, *journey*, 1.55)
MC	30	0-4	(*midday*, *noon*, 3.42)
			(*coast*, *forest*, 0.42)
WS	353	0-10	(*money*, *cash*, 9.08)
			(*football*, *tennis*, 6.63)
RW	2034	0-10	(*angrier*, *huffy*, 6.88)
			(*ulcerate*, *affect*, 2.40)
SCWS	2023	0-10	(*fear*, *panic*, 8.05)
			(*beam*, *shore*, 2.30)
MEN	3000	0-50	(*grass*, *lawn*, 48.00)
			(*beef*, *tomato*, 28.00)
SimLex	999	0-10	(*happy*, *glad*, 9.17)
			(*night*, *day*, 1.88)

Table 3.2: Benchmark datasets used to evaluate the word representations in the word similarity measurement task. Size (in number of pairs), similarity scales and examples.

3.4.1.2 Benchmark Datasets

To evaluate JointReps in the word similarity measurement task, we use multiple human-annotated word similarity benchmark datasets: Rubenstein-Goodenough (RG) [127], Miller-Charles (MC) [103], WordSim353 (WS) [55], Rare Words (RW) [94], Stanford's contextual word similarities (SCWS) [71], Marco, Elia and Nam (MEN) [27] and the SimLex-999 (SimLex) [66]. Each word-pair in these benchmark datasets has a manually assigned similarity score, which we consider as the gold standard rating for semantic similarity. Those benchmark datasets vary considerably in size from as small as 30 pairs to as large as 3000 pairs. Table 3.2 provides details of those datasets.

3.4.1.3 Results

In Table 3.3 we compare the word representations learnt by the JointReps for different semantic relations in the WordNet. All the word representations compared in Table 3.3 are with $d = 300$ dimensions. We use the Spearman's rank correlation coefficient as the evaluation measure between each word representation method ratings and the human ratings, and use Fisher transformation to test for statistical significance.

We use the WS dataset [55] as validation data to find the optimal value of λ for each relation type. Specifically, we minimise Equation 3.4 for different λ values, and use the learnt word representations to measure the cosine similarity for the word-pairs in the WS dataset. We then select the value of λ that gives the highest Spearman correlation with the human ratings on the WS dataset. This procedure is repeated separately with each semantic relation type R. We found that $\lambda = 10000$ to perform consistently well on all relation types. The level of performance if we had used only the corpus for learning word representations (without using a KB) is shown in Table 3.3 as the corpus only baseline. This baseline corresponds to setting $\lambda = 0$ in Equation 3.4.

From Table 3.3, we see that by incorporating most of the semantic relations found in the WordNet JointReps does improve over the corpus only baseline. This result supports our proposal to use both a corpus and a KB jointly for learning word representations. Among the relation types, synonym reports the best performance in RG, MC, SCWS and MEN whereas hypernym reports the best

Method	**Relation**	**RG**	**MC**	**RW**	**SCWS**	**MEN**	**SimLex**
corpus only	-	0.7545	0.6796	0.2522	0.4829	0.7015	0.3274
JointReps	Synonym	**0.7879**	**0.7614**	0.2674	**0.5103**	**0.7367***	0.3492
	Hypernym	0.7774	0.7330	0.2536	0.5034	0.7335*	**0.3576***
	Hyponym	0.7720	0.7193	0.2616	0.5040	0.7292*	0.3575
	Member-holonym	0.7655	0.6985	0.2536	0.4869	0.7059	0.3310
	Member-meronym	0.7613	0.6952	0.2537	0.4867	0.7070	0.3332
	Part-holonym	0.7740	0.7144	0.2682	0.4937	0.7220*	0.3298
	Part-meronym	0.7814	0.7338	**0.2714**	0.4980	0.7215*	0.3317

Table 3.3: Performance of the JointReps using different semantic relation types on the word similarity benchmark datasets.

performance in SimLex. The fact that word similarity benchmarks contain many word-pairs that are similar explains the effectiveness of synonymy. Moreover, part-meronyms, part-meronyms and synonyms perform well in predicting the semantic similarity between rare words (RW), is important because it shows that by incorporating a KB we can learn better word representations. This result is important because it shows that a KB can assist the representation learning of rare words, among which the co-occurrences are small even in large corpora [94].

3.4.2 Word Analogy Prediction

Mikolov et al. [102] proposed the word analogy prediction as a new task to evaluate the word vector representations. Since then, it became one of the most popular and widely used tasks for that purpose [9]. To understand the task, we first, in Section 3.4.2.1, describe the idea behind it and how it can be used to evaluate the word embeddings. Next, in Section 3.4.2.2, the benchmark datasets that are used with this task is presented. The evaluation results are then discussed in Section 3.4.2.3.

3.4.2.1 Task Description

The word analogy prediction task is based on the idea that the vector difference between embeddings for two words can be used to represent the relationship between those words. Consequently, prior work on word vector representations learning has evaluated the accuracy of the trained word representations by using them to solve word analogy problems.

Specifically, given the words a, b, and c, the task here is to predict the word d that fits best into the proportional analogy $a : b :: c : d$. For example, in the proportional analogy *England:London :: Japan:?*, we will need to predict a word that fits best as an answer to this analogy question (the correct answer in this example is *Tokyo*). Mikolov et al. [102] found interesting linguistic regularities implicitly learnt in the vector space that can be explored by the vector difference, hence able to solve such analogy questions. For example:

$$\boldsymbol{a}_{England} - \boldsymbol{b}_{London} \approx \boldsymbol{c}_{Japan} - \boldsymbol{d}_{Tokyo} \tag{3.9}$$

Here $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$ and $\boldsymbol{d}$ are the vectors of the words *England*, *London*, *Japan* and *Tokyo* respectively. Therefore, to answer an analogy question $a : b :: c :?$, we compute the cosine similarity between the $\boldsymbol{b} - \boldsymbol{a} + \boldsymbol{c}$ and each candidate word $\boldsymbol{d}$, and select the most similar candidate word as the predicted answer to the analogy question.

Dataset	**Size**	**Analogy Type**	**Examples**
Sem	8869	country-capital man-woman country-currency	*england:london::japan:tokyo* *king:queen::man:women* *india:rupee::mexico:peso*
Syn	10675	adjective-adverb comparative superlative	*amazing:amazingly::safe:safely* *cold:colder::old:older* *fast:fastest::tall:tallest*

Table 3.4: Benchmark datasets used to evaluate the word representations in the word analogy prediction task. Size (in number of analogy questions) and *three* selected analogy types from each dataset.

3.4.2.2 Benchmark Datasets

To evaluate JointReps in the word analogy prediction task, we used the Google word analogy dataset [101]. The Google dataset consists of two categories of analogy: Semantic (Sem) and Syntactic (Syn). Each category has a number of analogy types. In particular, Sem contains *five* types of semantic analogies and Syn has *nine* syntactic analogy types. Table 3.4 provides details of these datasets.

3.4.2.3 Results

In Table 3.5, we compare the accuracy performance of the JointReps against the corpus only baseline on answering the analogy questions in Sem and Syn datasets. We study the effect of using different WordNet semantic relations as the default relation type for the KB. We use the binomial exact test with Clopper-Pearson [33] confidence interval to test for the statistical significance.

Method	**Relation**	**Sem**	**Syn**
corpus only	-	58.94	65.46
JointReps	Synonym	59.90	71.02*
	Hypernym	**60.15***	**71.91***
	Hyponym	60.05*	70.75*
	Member-holonym	59.53	65.91
	Member-meronym	58.94	65.68
	Part-holonym	59.10	67.86*
	Part-meronym	59.36	67.65*

Table 3.5: Performance of the JointReps with different semantic relation types on the word analogy benchmark datasets.

From Table 3.5, we see that by jointly learning with a KB, we can always outperform the corpus only baseline, irrespective of the relation type. All the relations were effective for answering Sem and Syn analogy questions in the dataset. Among the relation types, hypernymy and synonymy report the best performance. This result is important because it shows that a KB can assist in learning word representations capable of capturing the syntactic and semantic meaning. Once more, the fact that JointReps could significantly improve performance on this task empirically justifies our proposal for using a KB in the word representation learning process.

3.4.3 Comparisons Against Prior Work

In Tables 3.6 and 3.7, we compare the JointReps in word similarity measurement and word analogy prediction tasks against previously proposed word representations learning methods that use both a corpus and a KB. Specifically, we compare

Method	RG	MC	RW	SCWS	MEN	SimLex
RCM	0.471	-	-	-	0.501	-
RC-NET	-	-	-	-	-	-
Retro (CBOW)	0.577	0.5693	0.2512	0.4764	0.605	0.2718
Retro (Skipgram)	0.745	0.7446	0.2498	0.4813	0.657	0.3911
Retro (corpus only)	0.7865	0.7544	0.2552	0.4802	0.673	**0.3936**
JoinReps (synonym)	**0.7879**	**0.7614**	**0.2674**	**0.5103**	**0.7367***	0.3492

Table 3.6: Comparisons of JointReps against prior work on word similarity measurement task.

Method	Sem	Syn
RCM	-	29.90
RC-NET	34.36	44.42
Retro (CBOW)	36.65	52.50
Retro (Skipgram)	45.29	65.65
Retro (corpus only)	**61.11**	68.14
JointReps (synonym)	59.90	**71.02***

Table 3.7: Comparisons of JointReps against prior work on word analogy prediction task.

against Relation Constraint Model (RCM) [148], Relational and Categorical framework (RC-NET) [145], and Retrofitting (Retro) [51]. Details of those methods were provided in Section 2.3.3.1.

In both tables (Tables 3.6 and 3.7), we use the publicly available source codes of Retro to retrofit the vectors learnt by CBOW (Retro (CBOW)), and Skipgram (Retro (Skipgram)). We also retrofit the vectors learnt by the corpus only baseline (Retro (corpus only)). All of the above-mentioned methods are trained using ukWaC as the corpus and synonyms are extracted from the WordNet as the KB. Unfortunately, the implementations or the trained word representations were not available for the RCM and the RC-NET methods. Therefore, for those methods, we compare the results reported in the original publications. A dash in Tables 3.6 and 3.7 indicates that the performance on that dataset was not reported in the original publication.

From Table 3.6, we see that the JointReps reports the best performance on most of the word similarity measurement benchmark datasets. Particularly, JointReps obtains the best score on RG, MC, RW, SCWS, and MEN, whereas Retro (corpus only) reports the best results on the SimLex datasets. A similar observation was also found in the word analogy prediction task. In Table 3.7, JointReps reports the best performance in the Syn benchmark dataset statistically significantly outperforming all the prior work. Moreover, in the Sem dataset, JointReps obtains a statically significantly better performance over RC-NET, Retro (CBOW) and Retro (Skipgram) but not Retro (corpus only).

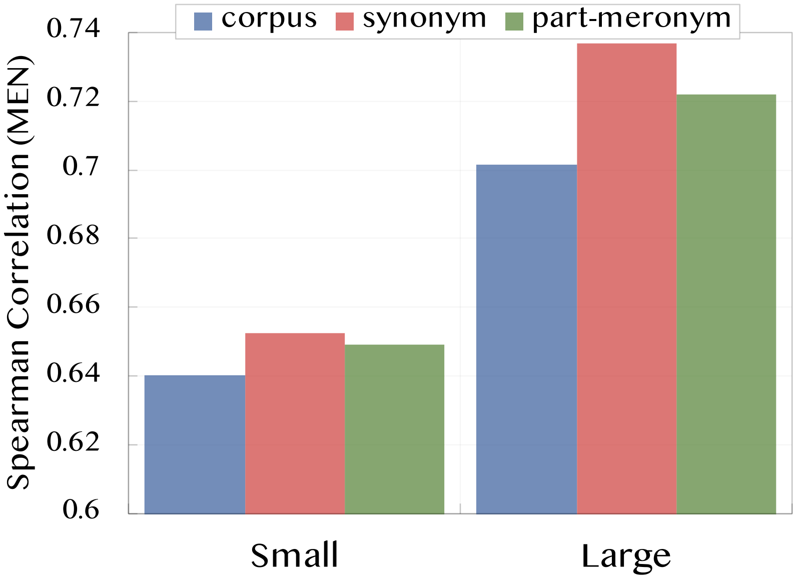


Figure 3.1: The effect of the corpus size (small and large) on JointReps, evaluated on the word similarity measurement task using MEN dataset.

3.4.4 Effect of Corpus Size

To evaluate the effect of the corpus size on the performance of the JointReps, we select a random subset containing 20% of the sentences in the ukWaC corpus, which we call the small corpus, as opposed to the original large corpus. In Figure 3.1, we compare three settings: (i) corpus (corresponds to the baseline method for learning using only the corpus, without the KB), (ii) synonym (JointReps method with synonym relation) and (iii) part-meronyms (JointReps method with part-meronym relation). Figure 3.1 shows the Spearman correlation coefficients on the MEN benchmark dataset for the word similarity measurement task. We see that in both small and large corpora settings, we can improve upon the corpus only baseline by incorporating semantic relations from the KB. Similar trends were observed for the other relation types as well.

In the next chapter (Section 4.6.5), we look into this in more details. Specifically, we conduct a comprehensive experiment with a wide range of different corpora sizes to further investigate the effect of the corpus size on the proposed JointReps.

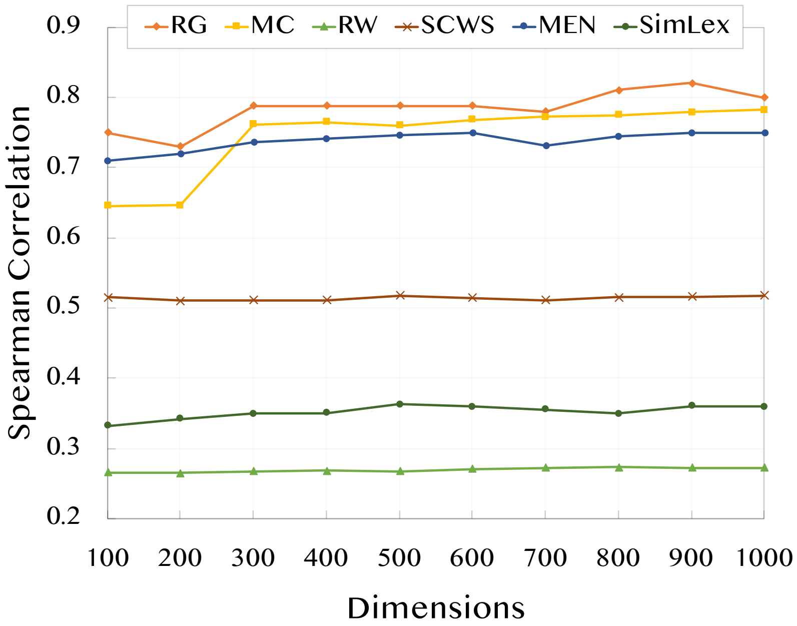


Figure 3.2: The effect of the dimensionality of the word representations learnt by JointReps using the synonym relation, evaluated on word similarity benchmark datasets.

3.4.5 Effect of Dimensionality

We evaluate the effect of the dimensionality d on the word representations learnt by the JointReps. In Figure 3.2 we report results when we use the synonym relation in the JointReps and evaluate on the word similarity benchmark datasets. Similar trends were observed for the other relation types and benchmarks. From Figure 3.2 we see that the performance of the JointReps is relatively stable across a wide range of dimensionalities. In particular, with as less as 100 dimensions, we can obtain a level of performance that outperforms the corpus only baseline. On RG, MC, and MEN benchmark datasets, we initially see a gradual increase in performance with the dimensionality of the word representations. However, this improvement saturates after 300 dimensions, which indicates that it is sufficient to consider 300-dimensional word representations in most cases. More importantly, adding new dimensions does not result in any decrease in performance.

3.5 Summary

In this chapter, a *Joint Representation Learning for Additional Evidence* (JointReps) method was proposed. It uses the information available in a KB as *additional evidence* alongside the co-occurrence distribution available in the corpus to improve the learnt word vector representations. For this purpose, JointReps uses the corpus to define a learning count- and prediction-based objective subject to the constraints derived from the KB. Experiments using ukWaC as the corpus and WordNet as the KB show that we can significantly improve word representations learnt using only the corpus by incorporating the information from the KB. Moreover, the proposed JointReps significantly outperforms previously proposed methods for learning word representations using both a corpus and a KB in both a word similarity measurement and a word analogy prediction tasks on a range of benchmark datasets. It was also shown that the effectiveness of using the KB with the corpus is prominent irrespective of the corpus size. The chapter concluded with a study showing that the performance of the proposed JointReps is stable over a wide-range of dimensionalities of word representations.

In the next chapter, we will present an extension to JointReps that attempts to enhance the incorporation of the KB information into the word representations learning process. In particular, since JointReps uses the KB information to derive relational constraints for each co-occurring word-pair in the corpus to guide the optimisation process during the word representations learning phase, it is likely that a large fraction of the words in the corpus might not be covered considering the fact that corpora are typically much larger than the KBs. Therefore, two new methods will be proposed that expand the KB from the corpus information to overcome this problem.

Chapter 4

Dynamic Knowledge Base Expansion

4.1 Introduction

In the previous chapter, we proposed the JointReps method, which uses the KB information as additional evidence to the corpus co-occurrence information for learning word representations. For each word-pair co-occurring in the corpus, JointReps attempts to enhance the learnt word representations by using the semantic relations between those words that exist in the KB as constraints during the learning process. In doing so, JointReps demonstrated a significant improvement in the learnt word representations in various tasks outperforming the corpus only baseline approach and prior work that use both the corpus and the KB for learning word representations.

However, in practice, a KB might not contain all the words in the corpus. Because in JointReps we derive constraints only from the KB, the coverage of the constraints derived from the KB might cover only a small fraction of the words in the corpus. In fact, for example, the vocabulary size in the ukWaC corpus (which we used as a corpus in JointReps) is $434,826$ whereas the vocabulary size in all the semantic relations extracted from the WordNet (which we used as a KB in JointReps) is $52,002$, which is less than 1/8th.

To overcome this problem, in this chapter, we enhance the incorporation process of the KB in the JointReps by proposing two novel expansion methods:

(i) Nearest Neighbour Expansion (NNE) and (ii) Hedged Nearest Neighbour Expansion (HNE), that expand the KB using the information extracted from the corpus. In NNE, we expand and dynamically update the KB by extracting features from the corpus based on co-occurrence counts, instead of considering only the original information in the KB. HNE works in a fashion similar to NNE, but more robustly by filtering co-occurrence noise in the corpus prior to dynamically updating the KB. Both NNE and HNE expand the KB dynamically, considering the word representations learnt, to improve the coverage and accuracy of the word representations. Because the expansion of the KB happens at run time, we call it a *dynamic* expansion. It is noteworthy that the purpose of performing this expansion is to derive constraints that guide the optimisation process and not to build better KBs.

The rest of this chapter is structured as follows. In Section 4.2 we revisit the method presented in the previous chapter to incorporate the KB in JointReps. Next, in Sections 4.3 and 4.4, we present the newly proposed dynamic expansion methods NNE and HNE respectively. Then, in Section 4.6 we provide an extensive evaluation of the NNE and HNE. Finally, Section 4.7 summarises the material presented in this chapter.

4.2 Static Knowledge Base (SKB)

To incorporate the KB into the learning process of JointReps in the previous chapter, we extract relations from the KB and use them to enforce the constraints that if two words co-occur together in the corpus and have a semantic relation in the KB, then they must have similar word representations. We only used the data that originally existed in the KB without any KB expansion. That is, the number of edges of each semantic relation extracted from WordNet (given by Table 3.1) were the only constraints derived from the KB that contributed towards the learning process. For example, from Table 3.1, we see that it contains $87,060$ pairs of synonyms extracted from WordNet, we considered only those pairs in our KB objective given by Equation 3.3 during the learning process, without applying any expansion on those synonyms pairs.

Henceforth, we refer to this method of incorporating the KB into JointReps that was presented in Chapter 3 as Static Knowledge Base (SKB) and consider it to be a baseline method for comparing against the new expansion methods we describe in the following sections. In SKB, we assume the relation strength function $R(w_i, w_j)$ given by Equation 3.3 to be a binary function that returns 1 if there exists a semantic relation R between the two words w_i and w_j in the KB $\mathcal{S}$ and 0 otherwise. Because SKB does *not* dynamically expand the KB but uses the original data available in the KB, we refer to it as a *static*.

4.3 Nearest Neighbour Expansion (NNE)

Typically, a corpus would cover a much larger vocabulary, and more relations can be derived from it as compared to that by a KB. If we can somehow use the information extracted from the corpus to expand the KB dynamically, then we can derive more constrains for the joint optimisation process, thereby making better use of the KB in the JointReps.

Let us consider a KB where knowledge is represented in the form of relational tuples (u, R, v), involving a relation R that exists between two words u and v. In what follows, we denote the set of vertices (vocabulary) in the KB by $\mathcal{N}$, and its set of relational tuples by $\mathcal{E}$. If two words $u, v \in \mathcal{N}$ have a relation R, then we have $(u, R, v) \in \mathcal{E}$.

In NNE, we assume that if two words u and v frequently co-occur in a corpus, then it is likely that there exists some semantic relation between those two words. We can compute the strength of association between two words using their co-occurrence count in the corpus, to create a k-nearest neighbour (K-NN) graph where u is connected to v if and only if v is among the top-k nearest neighbours of u. In our experiments, we use the Positive Pointwise Mutual Information (PPMI) [115] as the association measure, and selected the top-K neighbours according to the highest PPMI values between two words. Denoting the co-occurrence count between u and v in the corpus by $c(u, v)$ and the occurrence of u and v respectively by $c(u, *)$ and $c(*, v)$, the PPMI between u and v, $\text{PPMI}(u, v)$ is computed as follows:

$$\text{PPMI}(u, v) = \max\left(\log\left(\frac{c(u, v)c(*, *)}{c(u, *)c(*, v)}\right), 0\right) \tag{4.1}$$

Let us denote the set of k nearest neighbours of u in the corpus by $K_{NN}(u)$.

Between a word u that occurs in the KB and a word v that only occurs in the corpus, if v is the nearest neighbour of u (i.e. $v \in k_{NN}(u)$), then we add v to the KB. Moreover, the relation between v and u is set to the default semantic relation of the KB. We call this dynamic expansion method as the Nearest Neighbour Expansion (NNE), and show its pseudo-code in Algorithm 2.

Algorithm 2 Nearest Neighbour Expansion (NNE).

Input: Word co-occurrence matrix X specifying the co-occurrences between words in the corpus $\mathcal{C}$, a KB $\mathcal{S} = (\mathcal{N}, \mathcal{E})$ with a vocabulary $\mathcal{N}$ and a set of relational tuples $\mathcal{E}$, hyperparameter K specifying the number of nearest neighbours (NN) to consider.

Output: $\mathcal{S} = (\mathcal{N}, \mathcal{E})$

```
for $v \in \mathcal{C}$ do
    if $\exists u \in \mathcal{N}$ s.t. $v \notin \mathcal{N} \wedge v \in K_{NN}(u)$ then
        $\mathcal{E} \leftarrow \mathcal{E} \cup \{(u, R, v)\}$
        $\mathcal{N} \leftarrow \mathcal{N} \cup \{v\}$
    end if
end for
return $\mathcal{S}$.
```

4.4 Hedged Nearest Neighbour Expansion (HNE)

One drawback of the NNE method described in the previous section is that it considers the neighbourhood $K_{NN}(u)$ of each word u currently in the KB independently of the other neighbours when deciding whether a new word v should be linked to u. This operation can be problematic due to two reasons. First, some hub words that are associated with more than one word such as *everything* are not suitable as expansion candidates because they lack specificity. PPMI does not necessarily overcome the hubness problem [43]. Second, some words can be ambiguous, and if we expand each word individually as done by NNE, we might incorrectly link different senses of a word from the corpus. For example, let us assume that *Apple* and *Microsoft* are connected via *competitor* relation in a KB. Moreover, let us assume that *banana* co-occurs highly with *Apple* in the corpus. Because we do not assume the corpus to be sense annotated, we might incorrectly link *banana* to *Apple* because it is the nearest neighbour of the fruit sense of *Apple* in the corpus.

We propose two modifications to the NNE method to overcome the above-mentioned disfluencies. First, we require a word v to be the nearest neighbour of two words u and h that are already in the KB before we consider v to be an expansion candidate for the KB. This requirement will reduce the attachment of noisy co-occurrences. Second, we require some semantic relations to exist between u and h in the KB before we consider v to be an expansion candidate for the KB. In our previous example, *banana* (h) is unlikely to co-occur a lot with *Microsoft* (u) in the corpus, therefore *banana* will not be considered as an expansion candidate of *Apple*. Because of stricter neighbourhood requirement of this method that limits the extent of the expansion, we call it the Hedged Nearest Neighbour Expansion (HNE) method. The pseudo code for HNE is shown in Algorithm 3.

Algorithm 3 Hedged Nearest Neighbour Expansion (HNE).

Input: Word co-occurrence matrix X specifying the co-occurrences between words in the corpus $\mathcal{C}$, a KB $\mathcal{S} = (\mathcal{N}, \mathcal{E})$ with a vocabulary $\mathcal{N}$ and a set of relational tuples $\mathcal{E}$, hyperparameter K specifying the number of nearest neighbours (NN) to consider.

Output: $\mathcal{N}$ (Expanded $\mathcal{S}$)

```
$\mathcal{S} = (\mathcal{N}, \mathcal{E})$
for $v \in \mathcal{C}$ do
    if $\exists u, h \in \mathcal{N}, v \notin \mathcal{N}$ s.t. $v \in K_{NN}(u) \wedge v \in K_{NN}(h) \wedge (u, R, h) \in \mathcal{E}$ then
        $\mathcal{E} \leftarrow \mathcal{E} \cup \{(u, R, v), (h, R, v)\}$
        $\mathcal{N} \leftarrow \mathcal{N} \cup \{v\}$
    end if
end for
return $\mathcal{S}$.
```

4.5 Relational Strength in the Expanded KB

Considering that the nearest neighbours used to expand the KB in both NNE and HNE are found from the corpus purely based on co-occurrence statistics, they might not actually be reflecting the same semantic relation as in the KB. Moreover, PPMI values computed from sparse co-occurrences can be unreliable. In contrast, the KB might be a cleaner and an accurate semantic resource that is manually created and maintained. Therefore, we must impose a higher level of confidence on the original words and relations described in the KB (before the expansion) than the candidates we automatically append (either by NNE or HNE) from the corpus.

To prioritise the words that originally appeared in the KB over the automatically added words from the corpus, we set the relational strength $R(u, w)$ for two words u and w that appeared in the KB prior to dynamic expansion to 1. Meanwhile, the relational strength $R(u, v)$ for a word u that originally appeared in the KB and an expansion candidate v selected from the corpus is set to the normalised PPMI value between u and v, where we normalise the PPMI values by the sum of PPMI values over all k nearest neighbours. Specifically, the relational strength $R(u, v)$ for two words in the KB after the dynamic expansion process is defined as follows:

$$R(u,v) = \begin{cases} 1, & (u,R,v) \in \mathcal{E} \\ \\ \frac{\mathrm{PPMI}(u,v)}{\sum_{v' \in K_{NN}(u)} \mathrm{PPMI}(u,v')}, & u \in \mathcal{N} \wedge v \in K_{NN}(u) \text{ OR} \\ & u, h \in \mathcal{N}, v \in K_{NN}(u) \wedge v \in K_{NN}(h) \wedge (u,R,h) \in \mathcal{E} \end{cases} \tag{4.2}$$

4.6 Evaluation and Results

To firstly study the impact of the expansion methods NNE and HNE on the KB data, Sections 4.6.1 and 4.6.2 respectively provide a quantitative and qualitative analysis on the KB before and after the expansion applied. Following that, in Section 4.6.3, we study the impact of NNE and HNE on the learnt word representations by comparing the performance of JointReps with NNE and HNE against the SKB in the word similarity and word analogy tasks.

To learn the word representations, we follow the same experimental settings discussed in Section 3.3 and used to train JointReps in the previous chapter. In all the upcoming results tables of the word similarity and word analogy tasks the reported values are the Spearman correlation coefficients and accuracy performance for the word similarity and word analogy datasets respectively. The best performance in these tables for each task is shown in **bold** and statistical significance is indicated by asterisk (*). Next, in Section 4.6.4, we observe the impact of multi-rounds of expansions using NNE and HNE. Finally, Section 4.6.5 and 4.6.6 respectively provide an extensive analysis of the impact of the corpus and KB sizes on the performance of JointReps with SKB, NNE and HNE.

4.6.1 Expansion Effect on the KB

To study the impact of expanding the KB using NNE and HNE on the KB size, we compare the size of each semantic relation type in WordNet before and after the expansion. To tune the neighbourhood size K in NNE and HNE, we use the same validation data and settings previously used in Section 3.4.1.3 to tune λ in Equation 3.4. We found that $K = 5$ to perform consistently well for all semantic relation types.

Relation	**Edges**		
Type	**SKB**	**NNE**	**HNE**
Synonyms	87,060	108,510	104,123
Antonyms	4,064	7,004	5,325
Hypernyms	119,029	144,199	138,922
Hyponyms	122,926	141,961	138,010
Member-holonyms	11,506	13,716	12,033
Member-meronyms	11,431	12,706	11,651
Part-holonyms	13,082	18,222	16,557
Part-meronyms	13,251	18,191	16,186

Table 4.1: KB size (in number of edges) for different relation type under different expansion methods with $K = 5$ expansion words. The SKB column denotes the number of edges originally existed in WordNet before the expansion.

Table 4.1 shows the number of tuples extracted for relation type (SKB) and the size of the KB after expanding with the corpus using NNE and HNE methods. From Table 4.1, we can see that on average each relation type has expanded by about 25% and 20% using either NNE or HNE respectively. Because of the extra requirements imposed by HNE over NNE, HNE is expected to assign fewer number of expansion candidates than NNE.

Word	**Associated Words**		
	SKB	**NNE**	**HNE**
autopilot	autopilots	everything land copilot assist american	copilot assist software
imagination	imagery resource resourcefulness imaging imaginativeness vision	art originality mind unfettered fascinate	sight picture sense
pineapple	ananas	soon red pineapples pecan mango	fruit pineapples flowers
magyar	hungarian	culture group central re english	romania language culture
china	cathay taiwan chinaware prc	amity europe beijing south shanghai	beijing shanghai bhutan
sulfur	sulphur	fire test oxide hydrogen reference	oxide odor oxygen

Table 4.2: Examples of KB expansion using NNE and HNE on synonym relation type. The SKB column denotes the associated synonym words originally existed in WordNet before the expansion.

4.6.2 Qualitative Analysis

To qualitatively understand the differences among the proposed NNE and HNE expansion methods, in Table 4.2 we show randomly selected examples from SKB, NNE and HNE. We can see from Table 4.2 that NNE and HNE associate some related words that were not originally available with the SKB. For instance, words like *copilot*, *pineapples* and *beijing* have been associated with *autopilot*, *pineapple* and *china* respectively using NNE and HNE. Moreover, Table 4.2 shows that HNE was able to successfully eliminate some potential noisy expansion words. For example, the words *everything*, *american*, *soon* and *amity* have been associated as expansion words with *autopilot*, *china* and *pineapple* respectively using NNE, but excluded by HNE. Furthermore, because we limit the expansion candidates to the top-K neighbours (K = 5), we can see in Table 4.2 that some words are included in HNE but not in NNE. In such cases, the top five neighbours, according to NNE, do not meet the HNE requirements.

4.6.3 Impact of Dynamic Expansion on JointReps

To compare the word representations learnt by JointReps using the two dynamic KB expansion methods NNE and HNE over SKB, we train word representations using each method separately. We then evaluate the word representations on the two standard tasks, word similarity and word analogy, which have been used to evaluate JointReps in the previous chapter. Details of these tasks and their benchmark datasets have been discussed in details in Sections 3.4.1 and 3.4.2.

Method	SCWS	MEN	SimLex	Sem	Syn
SKB	0.5103	0.7367	0.3492	**59.9**	71.02
NNE	**0.5128**	0.7390	**0.3535**	59.75	**71.25**
HNE	0.5122	**0.7409**	0.3515	59.75	71.02

Table 4.3: Comparisons of JointReps among SKB, NNE and HNE using synonym relation type on word similarity (SCWS, MEN and SimLex) and word analogy (Sem and Syn) benchmark datasets.

In Table 4.3, we compare the results that we obtained by expanding the KB using NNE and HNE in synonymy relation against the SKB in the word similarity and the word analogy tasks. For the word similarity benchmark datasets, we only report the result of the three selected datasets (SCWS, MEN, and SimLex). We selected SCWS, MEN and SimLex here because those datasets have the largest numbers of word-pairs among all the word similarity benchmark datasets. From Tables 4.3, we can see that both NNE and HNE outperforms SKB in most of the benchmarks. In particular, NNE reports the best performance in SCWS, SimLex and Syn, whereas the best score in MEN achieved by HNE. However, the differences between the three methods are not statistically significant after one expansion round. As we later discuss in Section 4.6.4, NNE and HNE significantly outperform SKB in various benchmarks when we repeat the expansion process multiple rounds.

4.6.4 Multi-Rounds of Expansion

The NNE (Algorithm 2) and HNE (Algorithm 3) methods can be repeatedly used to expand a KB using the word representations learnt from previous rounds. Specifically, once we have expanded the KB using either NNE or HNE, we run Algorithm 1 with the same settings T=20 and $\lambda = 10,000$ to learn word representations. Next, we use those word representations to find the nearest neighbours used in NNE and HNE. We then expand the KB using Algorithms 2 or 3. Because the word representations learnt after expanding the KB could be better than the original word representations, by using the newer word representations we can hope to find more nearest neighbours, thereby further expanding the KB. Similar to the experiment in tuning the values of λ in Section 3.4.1.3, we use the WS dataset as validation data for tuning the number of expanding rounds. We observed that 10 rounds were sufficient where with further expansion the performance start falling behind the SKB baseline. We also observed that 5 rounds represent, on average, the peak point for most of the benchmark datasets. In Table 4.4, we compare the results that we obtained by expanding the KB with 5 rounds of expansion in all the 7 different WordNet semantic relations against the SKB. From Table 4.4, we can see that both NNE and HNE outperforms SKB in most of the benchmarks irrespective of the relation types. In particular, NNE on synonyms, hypernyms, part-holonyms and part-meronyms reports the best performance on most of the benchmarks, whereas HNE works better on hyponyms,

Method	Relation	SCWS	MEN	SimLex	Sem	Syn
SKB	synonym	0.5103	0.7367	0.3492	59.9	71.02
NNE		**0.5281**	0.7434	**0.3651**	**66.24**	**39.52**
HNE		0.5198	**0.7436**	0.3627	**60.1**	71.36
SKB	hypernym	0.5034	0.7335	0.3576	**60.15**	71.91
NNE		**0.5162**	0.7372	**0.3647**	60.13	**72.71**
HNE		0.5122	**0.7385**	0.3633	60.14	72.63
SKB	hyponym	0.5040	0.7292	0.3575	60.05	70.75
NNE		0.5105	0.7318	0.3582	**60.22**	**70.83**
HNE		**0.5109**	**0.7336**	**0.3583**	62.2	70.79
SKB	member holonym	0.4869	0.7059	0.3310	59.53	65.91
NNE		0.4882	0.7072	**0.3368**	59.64	66.09
HNE		**0.4897**	**0.7096**	0.3339	**59.71**	**66.16**
SKB	member meronym	0.4867	0.7070	0.3332	58.94	65.68
NNE		**0.4895**	0.7092	**0.3355**	59.28	65.92
HNE		0.4891	**0.7093**	0.3354	**59.38**	**65.97**
SKB	part holonym	0.4937	0.7220	0.3298	59.10	67.86
NNE		**0.5019**	0.7266	**0.3325**	59.24	67.92
HNE		0.5002	**0.7269**	0.3316	**59.31**	**67.95**
SKB	part meronym	0.4980	0.7215	0.3317	59.36	67.65
NNE		**0.5028**	0.7237	0.3328	**59.64**	**67.97**
HNE		0.5016	**0.7252**	**0.3334**	59.45	67.75

Table 4.4: Comparisons of JointReps among SKB, NNE and HNE using different relation types with 5 expansion rounds word similarity (SCWS, MEN and SimLex) and word analogy (Sem and Syn) datasets.

member-holonyms and member-meronyms.

To readily understand the impact of the multi-rounds of expansion, in Figure 4.1, we plot the Spearman correlation coefficient on SCWS against the number of expansion rounds with NNE and HNE. The horizontal lines correspond to the corpus only and SKB methods which either do not use the KB or does not expand the KB. From Figure 4.1 we can see that for both NNE and HNE, the performance increases with the number of expansion rounds, until approximately the 9th round, where the performance saturates. Similar trends were observed in all benchmark datasets, where multi-round expansion improves performance over single-round expansion in all cases but the performance either saturate or degrades because more noisy and irrelevant expansion candidates are introduced

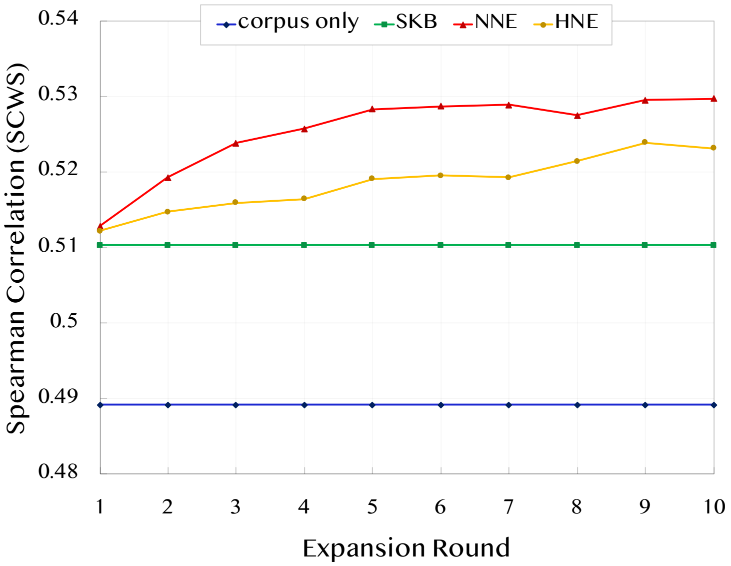


Figure 4.1: The impact of multi-rounds of expansion using NNE and HNE with synonym relation evaluated on the SCWS dataset.

in later expansion rounds.

% of ukWaC	Number of tokens	Size
100	2B	XL
70	1.4B	L
40	800M	M
20	400M	S
10	200M	XS

Table 4.5: Sub-corpora selected from ukWaC.

4.6.5 Impact of Corpus Size

In the previous chapter (Section 3.4.4), we have briefly studied the effect of the corpus size on the performance of JointReps. Here, we further thoroughly investigate the impact of the corpus size on JointReps with SKB and the new expansion methods, NNE and HNE. For this purpose, we use the five sub-corpora from the ukWaC corpus defined in Table 4.5, and train word representations with the complete WordNet KB. We evaluate the trained word representations using the word similarity and analogy benchmark datasets and report results in Table 4.6. Overall, as prior work has shown [119][106], Table 4.6 shows that a larger corpus size helps for obtaining a better level of performance. All the results reported in Table 4.6 use the synonym relation. From Table 4.6, we see that by incorporating the synonym semantic relation using SKB, NNE and HNE with different corpus sizes, the proposed method always outperforms the corpus only baseline on all benchmark datasets. Moreover, we see that NNE and HNE produce better

Method	Corpus Size	SCWS	MEN	SimLex	Sem	Syn
corpus only	XL	0.4829	0.7015	0.3274	58.94	65.46
SKB		0.5103	0.7367*	0.3492	**59.90**	71.02*
NNE		**0.5128**	0.739*	**0.3535**	59.75	**71.25***
HNE		0.5122	**0.7409***	0.3515	59.75	71.02*
corpus only	L	0.4719	0.6950	0.3235	57.37	64.65
SKB		0.4986	0.7278*	0.3461	58.64*	68.94*
NNE		**0.5037**	0.7302*	**0.3504**	58.28	**69.10***
HNE		0.5017	**0.732***	0.3481	**58.68***	69.07*
corpus only	M	0.4687	0.6892	0.3157	51.71	62.79
SKB		0.4950	0.7187*	0.3327	52.41	65.48*
NNE		0.4966	0.7211*	**0.3366**	**52.54**	**65.55***
HNE		**0.4981**	**0.7226***	0.3342	52.36	65.59*
corpus only	S	0.4509	0.6704	0.2978	43.08	56.77
SKB		0.4740	0.6923*	0.3113	43.27	58.22*
NNE		**0.4765**	0.6943*	**0.3163**	43.38	**58.44***
HNE		0.4762	**0.6963***	0.3126	**43.42**	58.25*
corpus only	XS	0.4446	0.6404	0.2636	31.72	48.99
SKB		0.459	0.6565	0.2741	32.01	49.61
NNE		0.4622	0.6580	**0.2772**	**32.16**	**49.73**
HNE		**0.4624**	**0.6595**	0.2749	32.10	49.69

Table 4.6: Performance of JointReps using SKB, NNE and HNE against the corpus only baseline in various corpus sizes with synonym relation on word similarity (SCWS, MEN and SimLex) and word analogy (Sem and Syn) benchmark datasets. **Bold** indicates the best performance in each dataset, and * indicates the statistical significant against the corpus only baseline.

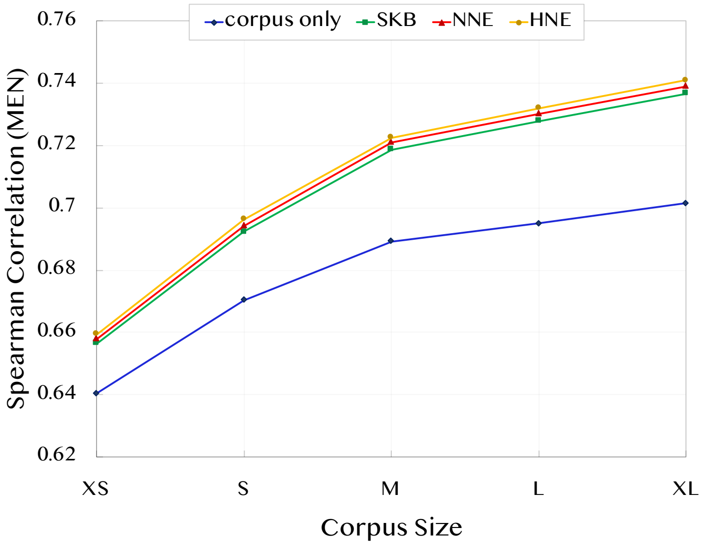


Figure 4.2: The effect of varying the size of the corpus under SKB, NNE, HNE on the MEN benchmark dataset.

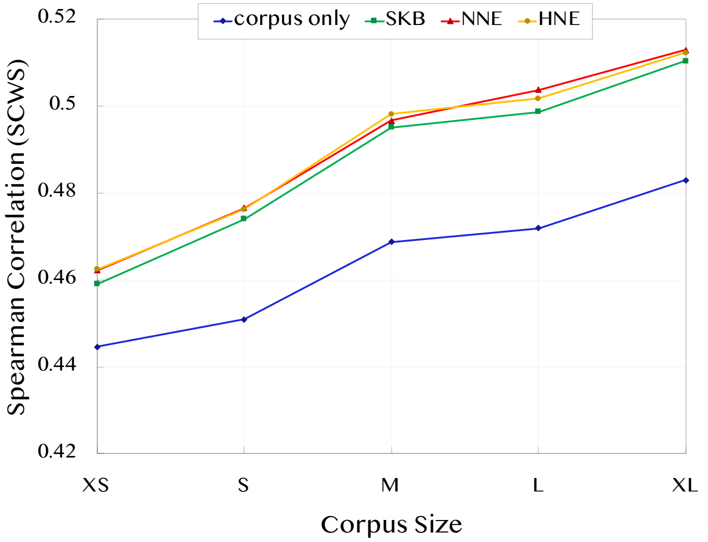


Figure 4.3: The effect of varying the size of the corpus under SKB, NNE, HNE on the SCWS benchmark dataset.

word representations over SKB in most of the benchmark datasets. In particular, NNE and HNE obtain a significant improvement over SKB for predicting similarity between words in SCWS and MEN benchmarks across all different corpus sizes. Moreover, in the word analogy prediction task, NNE and HNE constantly outperform SKB on Sem and Syn datasets, irrespective of the size of the corpus.

To readily understand the effect of the corpus size on the accuracy of the word representations learnt by JointReps using SKB, NNE and HNE, in Figures 4.2 and 4.3, we plot the Spearman correlation coefficients against the size of the corpus for respectively MEN and SCWS benchmark datasets.

% of synonym word-pairs	**Edges**			**Size**
	SKB	NNE	HNE	
100	87,060	108,510	104,123	XL
70	60,941	75,957	72,886	L
40	34,824	43,404	41,649	M
20	17,412	21,702	20,824	S
10	8,706	10,851	10,412	XS

Table 4.7: Different KB sizes (synonym relation) randomly selected from WordNet.

4.6.6 Impact of KB Size

To evaluate the impact of the size of the KB on JointReps, we randomly select pairs of synonyms from WordNet to create KBs of varying sizes as shown in Table 4.7. We jointly train with each KB and the entire ukWaC corpus. Figures 4.4 and 4.5 show the impact of varying the KB size on JointReps evaluated respectively on MEN and SCWS benchmarks. Similar trends were also observed with other benchmark datasets. We fixed the corpus size with XL and performed the experiments with various KB sizes. The horizontal lines in the two figures (Figures 4.4 and 4.5) correspond to the corpus-only baseline, which is unaffected when the corpus is not varied.

From Figures 4.4 and 4.5, we see that JointReps using SKB, NNE, and HNE continuously increase performance when we increase the size of the KB. This result suggests that we can still learn high-quality word representations by creating KBs with better coverage on top of what we can learn about word semantics from

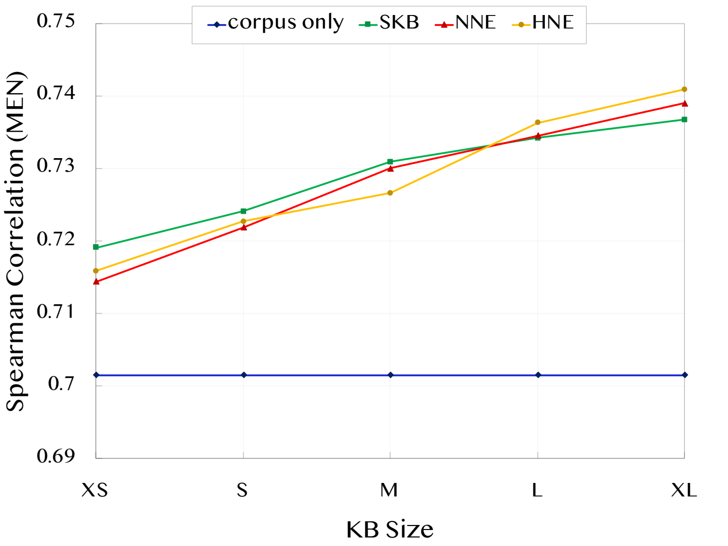


Figure 4.4: The effect of using different KB (synonym relation) sizes on JointReps with SKB, NNE and HNE evaluated on MEN benchmark dataset.

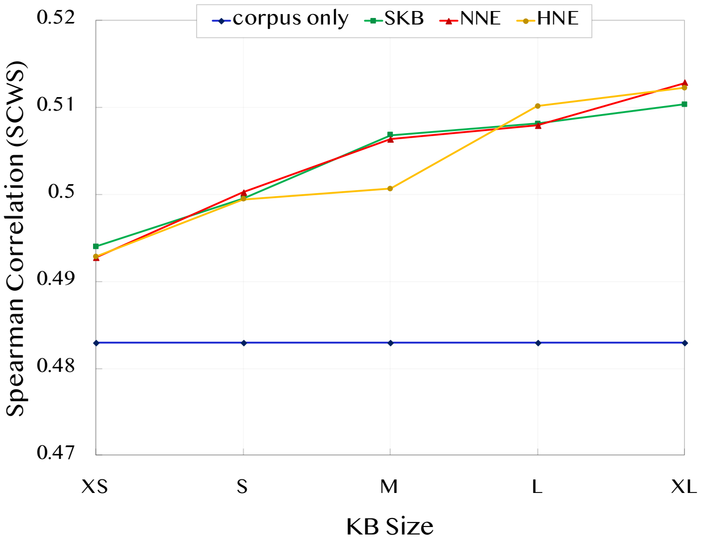


Figure 4.5: The effect of using different KB (synonym relation) sizes on JointReps with SKB, NNE and HNE evaluated on SCWS benchmark dataset.

large corpora. HNE, unlike NNE, requires expansion candidates to be mutual neighbours. With smaller KB, it is difficult to find such mutual neighbours, which results in HNE performing poorly compared to SKB and NNE. However, when we increase the size of the KB, HNE's performance increases. Moreover, although the performance gain is higher when the KB is large, we are still improving over the corpus only baseline with as small as 10% of the KB size. This result is important because it suggests that JointReps can assist the word representations learning for languages with limited availability of large-scale KBs.

4.7 Summary

This chapter has presented an enhancement of the incorporation process of the KB into the JointReps method. Specifically, we addressed the issue of the shortage of constraints that can be derived from the KB to cover the words in the corpus during the learning process of JointReps. For this purpose, Nearest Neighbour Expansion (NNE) and Hedged Nearest Neighbour Expansion (HNE) were proposed to expand the KB using the information extracted from the corpus. The experimental results on a range of benchmark datasets for semantic similarity measurement and word analogy prediction show that JointReps with NNE and HNE obtains improvements over the Static Knowledge Base (SKB) baseline on learning word representations. Moreover, we show that by repeatedly expanding the KB using NNE and HNE, we can further improve the word representations learnt by JointReps. Furthermore, extensive empirical experiments conducted with varying sizes of corpora and KBs show that JointReps with NNE and HNE reports consistent improvements over a wide range of different configurations of resources.

In the next chapter, the second proposed joint approach, concerning fine-tuning the word representations, is presented, namely the HWR approach. In HWR, a KB and a corpus are jointly utilised to learn fine-tuned word representations for hierarchical information.

Chapter 5

Fine-Tuning Word Representation for Hierarchical Information

5.1 Introduction

Organising the meanings of words in the form of a hierarchy is a standard practice ubiquitous in many fields such as linguistics [104], biology [44] and medicine [45]. Humans find it easier to understand a novel word (*a hyponym*) if its parent words (*hypernyms*) are already familiar to them [122]. For example, one can guess the meaning of the hyponym word *mallard* by knowing that the word *bird* is one of its hypernyms. Capturing such hierarchical information is vital for various NLP tasks such as question answering [72], taxonomy construction [109], textual entailment [35], text generation [19] and document clustering [57], among many others. Corpus-based word representations learning methods have shown some capability to encode hierarchical information in their learnt representations. However, these methods are not explicitly designed to learn word representations that encode hierarchical structure, thus empirically struggle in various other tasks as we see later in Section 5.3. Consequently, a new line of work, focusing on learning fine-tuned word representations for hierarchical information has emerged [111, 114, 140, 149].

So far in this thesis, we have proposed JointReps and its extension in Chapters 3 and 4 respectively, in which we have established the importance of learning word representations by combining a text corpus and a KB. Although in JointReps we have used several semantic relations including, *hypernym* and *hyponym* from a KB, we have *not* explicitly constructed the JointReps to transform the vector space to fine-tune it for a particular relation but to additionally evidence the distributional information found in the corpus. In other words, the word vector representations learnt by JointReps are not fine-tuned (not explicitly designed to encode a particular semantic relation) word representations but benefiting from several relations in the KB to emphasise the existing distributional information in the corpus.

In this chapter, we will instead look into transforming a vector space for fine-tuning it to encode the hierarchical structure that exists among words. Hence, we propose the second joint approach in this thesis for learning word vector representations from a corpus and a KB. Specifically, we propose the *Joint Hierarchical Word Representation* (HWR) method, in a specific order to encode the hierarchical structure of a KB in a vector space. To learn the word representations, the proposed HWR method considers not only the hypernym relations that exist between words in a KB but also the contextual information in a text corpus.

As noted earlier, some prior work has recently attempted to learn hierarchical word representations. However, several shortcomings are associated with such prior work (discussed in details in Section 2.3.3.2). For example, HyperVec [111] uses only a prediction-based objective, Skipgram, to incorporate the corpus-context into the learning process, which considers only local co-occurrences. Besides, HyperVec mainly focuses on pairwise hypernym relations between words in the KB, ignoring the full hierarchical path. The full hierarchical path of hypernyms not only gives a better understanding of the hierarchy than a single hypernym edge, but has been also empirically shown to be useful in various tasks as we see later in Section 5.3. Similar to HyperVec, LEAR [140] is limited to using pairwise hypernyms. Moreover, the Poincaré Ball model [114] only learns from the KB hypernyms, without any information from the corpus.

In this chapter, we intend to address the shortcomings of prior work discussed above. Firstly, we utilise the full hierarchical path of words from the KB, rather than only using pairwise hypernym relations. For example, to encode the hierarchical information of the word *bird* in Figure 5.1, we consider the full path (*bird*

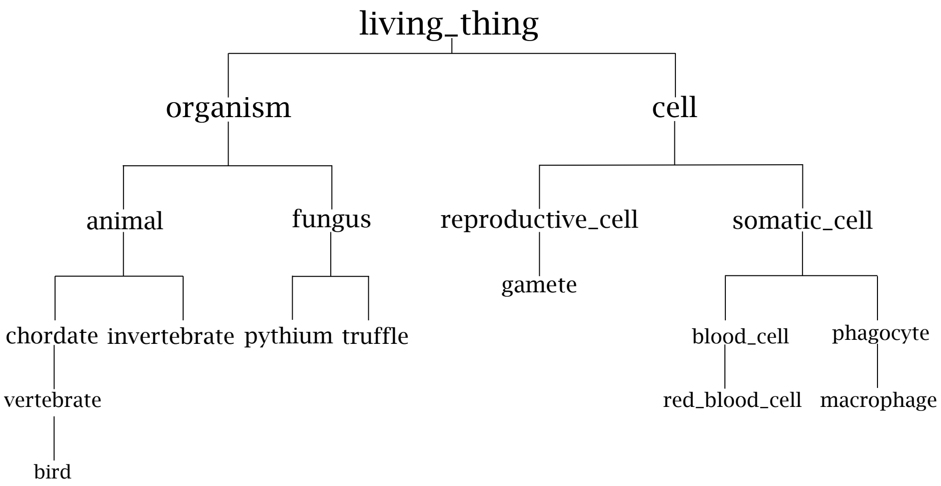


Figure 5.1: Example of hierarchy extracted from the KB (WordNet).

→ *vertebrate* → *chordate* → *animal* → *organism* → *living_ thing*) instead of only considering the pair (*bird*, *vertebrate*). Secondly, in HWR, we use a hybrid of count- and prediction-based objective to incorporate the corpus information, instead of purely relying on prediction-based approach. Moreover, we jointly learn the hierarchical word representations from the KB and the corpus enabling the proposed HWR to benefit from both resources.

The rest of this chapter is organised as follows. Section 5.2 presents the learning process of the proposed HWR. Section 5.3 then presents the experimental setup used to train HWR. An extensive evaluation of the HWR is then presented in Section 5.4. Finally, the chapter is concluded with a summary in Section 5.5.

5.2 Learning Process of HWR

We propose HWR, a method that learns word representations by encoding hierarchical structure among words in a KB and co-occurrence in a corpus. First, in Section 5.2.1, we introduce how we learn the hierarchical word representations from the KB. Followed by the incorporation process of the corpus into the learning process (Section 5.2.2). We then, in Sections 5.2.3, detail the *joint* learning method.

5.2.1 Learning HWRs from the KB

Given a hierarchical KB $\mathcal{T}$, we propose a method for learning d-dimensional HWR $\boldsymbol{w_i} \in \mathbb{R}^d$ for the each word $w_i \in \mathcal{V}$ in a vocabulary $\mathcal{V}$.

To explain the HWR learning method, let us revisit the example of *bird*'s hierarchical path from Figure 5.1, (*bird* → *vertebrate* → *chordate* → *animal* → *organism* → *living_ thing*) where the pairs *(bird,vertebrate)*, *(vertebrate,chordate)*, *(chordate,organism)* and *(animal,organism)* represent a *direct* hypernym relation, whereas *(bird, chordate)* and *(vertebrate, animal)* form an *indirect* hypernymic relation. We require our representations to encode not only the *direct* hypernym relations between a hypernym and its hyponyms, but also the *indirect* hypernymy.

We use a set of hierarchical paths, extracted from the taxonomy. Let us assume that w_i is a leaf node in the taxonomy and $\mathcal{P}(w_i)$ is the path that connects w_i to the root of the taxonomy. Because a taxonomy by definition arranges words in a hierarchical order, we would expect that some of the information contained in a leaf node w_i could be inferred from its parent nodes that fall along the paths $\mathcal{P}(w_i)$. Different compositional operators could then be used to infer the semantic representation for w_i using its parents, such as a recurrent neural network (RNN) [134]. However, for simplicity and computational efficiency, we learn the representation of a leaf node as the sum of its parents' representations. This idea can be formalised into an objective function $J_{\mathcal{T}}$ for the purpose of learning hierarchical word representations over the entire vocabulary $\mathcal{V}$ as follows:

$$J_{\mathcal{T}} = \frac{1}{2} \sum_{i \in \mathcal{V}} \left|\left| \boldsymbol{w}_i - \sum_{j \in \mathcal{P}(w_i)} \tilde{\boldsymbol{w}}_j \right|\right|_2^2 \tag{5.1}$$

The indirect hypernym at the top of a path (i.e. the root of a taxonomy) represents less (more abstract) information about w_i than its direct hypernym. In our previous example (*bird* → *vertebrate* → *chordate* → *animal* → *organism* → *living_ thing*), the direct hypernym *vertebrate* expresses more information about *bird* than the indirect hypernym *organism.* To reflect this we use a discounting term in (5.1) $\lambda(\tilde{w}_j)$ that assign a weight for each hypernym in the path as follows:

$$J_{\mathcal{T}} = \frac{1}{2} \sum_{i \in \mathcal{V}} \left|\left| \boldsymbol{w}_i - \sum_{j \in \mathcal{P}(w_i)} \lambda(\tilde{w}_j) \tilde{\boldsymbol{w}}_j \right|\right|_2^2 \tag{5.2}$$

Specifically, set $\lambda(\tilde{w}_j) = \exp(\mathcal{L}_{w_i} - \mathcal{D}_{\tilde{w}_j})$ where $\mathcal{L}_{w_i}$ and $\mathcal{D}_{\tilde{w}_j}$ respectively denote the length of the hierarchical path of the word w_i, and the distance measured in words between the word w_i and its hypernym $\tilde{w}_j$ in the path.

5.2.2 Incorporating the Corpus

We would like the proposed HWR to learn the hierarchical word representations from both the KB and the corpus. However, the objective function given by Equation (5.2) learns the word representations purely from the KB $\mathcal{T}$ and does not consider the information available in a corpus $\mathcal{C}$. To address this problem, we combine the corpus into the learning process by considering the co-occurrences between a hyponym and its hypernyms in the corpus. Specifically, for each hypernym $\tilde{w}_j$ that appears in the path of the hyponym w_i, we look up its co-occurrences in the corpus. For this purpose, we first create a co-occurrence matrix $\mathbf{X}$ between the hyponym and hypernym words within a context window in the corpus. The element X_{ij} of $\mathbf{X}$ denotes the total occurrences between the words w_i and $\tilde{w}_j$ in the corpus. We then use the GloVe objective given by Equation 3.1 to consider the co-occurrence between the hyponym word w_i and its hypernyms $\tilde{w}_j$ for the purpose of learning the representations.

5.2.3 Joint Objective

To formalise the final objective function of HWR, we would like to learn the hyponym $\boldsymbol{w}_i$ and hypernyms $\tilde{\boldsymbol{w}}_j$ representations that simultaneously minimise the KB-based objective given by Equation 5.2 and the corpus-based objective given by Equation 3.1. Thus, we combine the two objectives into a *joint* objective as follows:

$$J_{HWR} = J_{\mathcal{T}} + J_{\mathcal{C}} \tag{5.3}$$

To minimise the *joint* objective defined above (Equation 5.3) with respect to the parameters $\boldsymbol{w}_i$, $\tilde{\boldsymbol{w}}_j$, b_i and b_j, we compute the gradient of it with respect to those parameters. The gradients of $\boldsymbol{w}_i$ and $\tilde{\boldsymbol{w}}_j$ are computed as follows:

$$\frac{\partial J_{HWR}}{\partial \boldsymbol{w}_i} = \boldsymbol{w}_i - \sum_{j \in \mathcal{P}(w_i)} \lambda(\tilde{w}_j)\tilde{\boldsymbol{w}}_j + \sum_j f(X_{ij})\tilde{\boldsymbol{w}}_j\big({\boldsymbol{w}_i}^\top \boldsymbol{\tilde{w}_j} + b_i + b_j - \log(X_{ij})\big) \tag{5.4}$$

$$\frac{\partial J_{HWR}}{\partial \tilde{\boldsymbol{w}}_j} = -\lambda(\tilde{\boldsymbol{w}}_j)(\boldsymbol{w_i} - \sum_{j \in \mathcal{P}_(w_i)} \lambda(\tilde{\boldsymbol{w}}_j)\tilde{\boldsymbol{w}}_j) + \sum_i f(X_{ij})\boldsymbol{w}_i\big({\boldsymbol{w}_i}^\top \boldsymbol{\tilde{w}_j} + b_i + b_j - \log(X_{ij})\big) \tag{5.5}$$

Whereas the gradients of b_i and b_j are computed as defined previously by Equations 3.6 and 3.8.

5.3 Experiments

The experiment settings to train the proposed HWR method is provided in this section. The section begins with details of the applied pre-processing steps and the adopted KB training data (Section 5.3.1). That is followed by Section 5.3.2 where the model setup and the training details regarding the model's hyperparameters and the optimisation process are provided.

5.3.1 Training Data

We use the WordNet [104] as a KB in our experiments. However, it should be noted that any KB can be used as $\mathcal{T}$ with the proposed HWR provided that the hypernym relations that exist between words are specified. The average words' hierarchical path length in WordNet is 7. Following the recommendation in prior work on extracting taxonomic relations, we exclude the top-level hypernyms in each path. For example, Anh et al. [7] found that words such as *object*, *entity* and *whole* in the upper level of the hierarchical path to be too abstract and vague. Moreover, words such as *physical_ entity*, *abstraction*, *object* and *whole* appear in the hierarchical path of respectively 58%, 47.27%, 34.74% and 30.95% of the words in the WordNet. As such, we limit the number of words in each path to 5 hypernyms and obtained direct and indirect hypernym relations. After this filtering step, we select 59, 908 distinct hierarchical paths covering a vocabulary of

$|\mathcal{V}|= 80,673$. As the corpus $\mathcal{C}$, we used the ukWaC, which was previously detailed in Section 3.3.1.

Algorithm 4 HWR learning.

Input: Hierarchical paths $\mathcal{P}$ specifying the hypernym relations between the vocabulary words $\mathcal{V}$ in the KB (taxonomy) $\mathcal{T}$, word co-occurrence matrix $\mathbf{X}$ specifying the co-occurrences between hyponym and hypernym words in the corpus $\mathcal{C}$, dimensionality d of the word representations, and the maximum number of iterations T.

Output: Vector Representations $\boldsymbol{w}_i, \tilde{\boldsymbol{w}}_j \in \mathbb{R}^d$, of all words $w_i, w_j \in \mathcal{V}$.

```
Initialise word vectors $\boldsymbol{w}_i, \tilde{\boldsymbol{w}}_j \in \mathbb{R}^d$ randomly.
for $t = 1$ to $T$ do
  for $i \in \mathcal{V}$ do
    for $j \in \mathcal{P}(w_i)$ do
      for $(i, j) \in \mathbf{X}$ do
        Use Equation 5.4 to update $\boldsymbol{w}_i$
        Use Equation 3.6 to update $b_i$
        Use Equation 5.5 to update $\tilde{\boldsymbol{w}}_j$
        Use Equation 3.8 to update $\tilde{b}_j$
      end for
    end for
  end for
end for
return $\boldsymbol{w}_i, \tilde{\boldsymbol{w}}_j \quad \forall w_i, w_j \in \mathcal{V}$.
```

5.3.2 Model Setup and Training

The hierarchical paths extracted from the WordNet contain words that are unigrams, bigrams, trigrams and 4-grams such as *bird*, *living_ thing*, *red_ blood_ cell* and *first_ law_ of_ motion*. Because we use the corpus to look up the co-occurrences of the hyponym word and its hypernyms in the hierarchical paths, when we build the co-occurrence matrix $\mathbf{X}$ from the corpus we consider not only the unigrams, but also the n-grams variations. For example, if we found the sequence "*red blood cell*" in the corpus we would consider it as a trigram (*red_ blood_ cell*) word. For the hyperparameters (e.g. context window size and minimum word frequency) used to create the co-occurrence matrix $\mathbf{X}$, we follow the same settings as in Section 3.3.2. We use SGD as the optimisation method. The overall algorithm of the learning method HWR is listed in Algorithm 4. The word representations are randomly initialised to the uniform distribution in the range $[-1, +1]$ for each dimension separately. Experimentally, $T = 50$ iterations were found to be sufficient for the proposed method to converge to a solution. It takes around 30 minutes to train the model to learn 300 dimensional word representations on a Xeon 2.9GHz 32 core 512GB RAM machine.

To give a further intuitive explanation of the process of learning hierarchical word embeddings using the proposed method following the steps given by Algorithm 4, let us consider a concrete example. In Line 3, assume that the hyponym word w_i that we are interested in learning the hierarchical embedding for is *bird*. Line 4 will then consider every hypernym word w_j specified in *bird*'s hierarchical path (*vertebrate* → *chordate* → *animal* → *organism* → *living_thing*) extracted from the KB. The next step is then to look up the co-occurrence statistics between *bird* and each hypernym word in the corpus (Line 5). That is then followed by updating, in Lines 6-9, the corresponding vector representations of the hypernym word (*bird*), hypernym words (*vertebrate*, *chordate*, *animal*, *organism*, and *living_thing*) and the bias terms associated with each word. Therefore, the hierarchical representation of *bird* is learnt as the sum of its hypernyms' representations.

5.4 Evaluation and Results

We evaluate the word representations learnt by the proposed HWR on *five* main tasks: (i) a supervised hypernym detection (Section 5.4.1), (ii) graded lexical entailment (Section 5.4.2), (iii) unsupervised hypernym detection and directionality (Section 5.4.3) and the newly-proposed (iv) hierarchical path prediction (Section 5.4.4) and (v) word reconstruction (Section 5.4.5) tasks.

In all the above tasks, we compare the performance of the proposed HWR with various prior work on learning word representations. Specifically, we compare the HWR against: (i) corpus-based approaches: CBOW, Skipgram and GloVe (discussed in details in Section 2.3.1, Chapter 2), (ii) the first joint approach proposed in this thesis, JointReps, using hypernym relation (Chapter 3) and (iii) the methods that learn hierarchical word representations: HyperVec, Poincaré and LEAR which were discussed in details in Section 2.3.3.2 (Chapter 2) and briefly highlighted in the introduction of this chapter (Section 5.1).

For the fairness of the comparison, we used the same ukWaC corpus that is used with the proposed method to train all the prior methods using their publicly available implementations by the original authors for each method, except for Poincaré model, which we used the gensim implementation [123]. Similarly, we used WordNet to extract the hypernym relations with the prior methods. In all the experiments, we also follow the same settings used with the proposed method, and set the context window to 10 words to either side of the target word, and remove the words that appear less than 20 times in the corpus. We set the negative sampling rate to 5 for Skipgram and 10 for Poincaré following respectively [90] and [114]. We learn 300 dimensional word representations in all experiments.

Dataset	**#Instances**	**Ratio pos/neg**	**Examples**
Kotlerman	2,940	0.42	(*loan*, *fee*, −)
Bless	14,547	0.11	(*tuna*, *food*, +)
Baroni	2,770	0.98	(*snake*, *animal*, +)
Levy	12,602	0.08	(*dog*, *cat*, −)

Table 5.1: Benchmark datasets for the supervised hypernym identification task.

5.4.1 Supervised Hypernym Detection

Supervised hypernym detection is a standard task for evaluating the ability of word representations to detect hypernyms. As such, we evaluate the word representations learnt by the proposed HWR and prior work on this task. We first describe the task on Section 5.4.1.1. Then, in Section 5.4.1.2, we shed some lights on the benchmark datasets that are usually used in this task and conclude the section with the evaluation results (Section 5.4.1.3).

5.4.1.1 Task Description

The supervised hypernym detection is modelled as a binary classification problem, where a classifier is trained using word-pairs (x, y) labelled as *positive* (i.e. a hypernym relation exists between the x and y) or *negative* (otherwise). Each word in a word-pair is represented by its pre-trained word representation. Several operators have been proposed in prior work to represent the relation between two words using their word representations such as the vector *concatenation* [11], *difference* and *addition* [142]. In our preliminary experiments, we found that *concatenation* performed best for supervised hypernym identification, which we use as the preferred operator. To identify hypernyms in this task, we train a binary Support Vector Machine with a Radial Basis Function (RBF) kernel, with distance parameter $\gamma = 0.03125$ and the cost parameter $C = 8.0$ tuned using the validation split of the benchmark datasets discussed below.

5.4.1.2 Benchmark Datasets

For the supervised hypernym detection task, we selected *four* widely used benchmark datasets: Kotlerman [79], Bless [14], Baroni [11] and Levy [88]. Table 5.1 provides details of those benchmark datasets. Each dataset contains word-pairs holding different semantic relations, in which the hypernym pairs are labelled with *positive* and any other relations such as *synonym*, *meronym* and *antonym* are labelled with *negative*. To avoid any *lexical memorisation*, where the classifier simply *memorises* the prototypical hypernyms rather than *learning* the relation, Levy et al. [91] introduced a disjoint version with no lexical overlap between the train (75%), test (25%) and validation (5%) splits for each of the above datasets, which we use for our evaluations.

5.4.1.3 Results

Table 5.2 shows the performance of different word representations learning methods using F1 and the Area Under the Receiver Operating Characteristic (ROC) Curve (AUC). Sanchez and Riedel [128] argued that AUC is more appropriate as an evaluation measure for this task because some of the benchmark datasets are unbalanced in terms of the number of positive vs. negative test instances they contain. We observe that the proposed HWR reports the best scores in two of the benchmark datasets. In the Levy dataset, HWR reports the best performance with a slight improvement over the other methods. Similarly, HWR scores the highest in the Baroni dataset where we can observe a strong difference between the hierarchical word representations methods (third category in the table) and other methods. In particular, HyperVec, LEAR and HWR significantly (binomial test, $p < 0.05$) outperform the other methods and HWR reports the best score in this dataset. This result is particularly noteworthy, because a prior extensive analysis of different benchmark datasets for supervised hypernym identification by Sanchez and Riedel [128] concluded that the Baroni dataset is the most appropriate dataset for robustly evaluating hypernym identification methods. These results empirically justify our proposal to use the full hierarchical path in a taxonomy, instead of merely a pairwise hypernym relation, for learning better hierarchical word representations.

However, Table 5.2 shows that even the methods that were trained only with a text corpus not specifically designed to capture a hierarchy, performed well with respect to the Bless and Kotlerman datasets, reporting a better or comparable performance to the hierarchical representations. For example, using the Bless dataset, LEAR reports the best performance but with a slight improvement over GloVe. Whereas in Kotlerman, GloVe reports the best performance among all the other methods. This particular observation aligns with Sanchez and Riedel's [128] conclusion of the incapability of such benchmark datasets, apart from Baroni, to capture hypernyms from word representations in such tasks.

Method	Bless		Baroni		Kotlerman		Levy	
	F1	AUC	F1	AUC	F1	AUC	F1	AUC
CBOW	88.41	87.43	67.84	68.30	53.79	54.72	67.41	67.47
Skipgram	87.47	86.29	67.66	68.04	56.77	57.11	70.98	68.13
GloVe	91.85	93.28	68.87	69.33	**57.61**	**57.72**	68.47	69.78
JointReps	89.86	88.94	68.95	69.48	54.76	55.38	67.60	68.06
HyperVec	86.56	82.78	73.82	74.26	54.30	55.51	57.63	57.78
LEAR	**92.84**	**93.98**	74.63	74.47	57.53	57.24	70.96	75.23
Poincaré	66.96	80.61	63.97	64.84	53.49	56.27	52.22	61.85
HWR	88.19	90.23	**74.72**	**75.03**	55.95	57.55	**71.92**	**76.66**

Table 5.2: Classifier (SVM) performance using the word representations learnt by the proposed method HWR and others methods as features in the supervised hypernym identification task on several benchmark datasets. Details of the classifier parameters are provided in Section 5.4.1.1.

5.4.2 Graded Lexical Entailment

As an alternative to relying on the binary decision used in the supervised hypernym detection task discussed in the previous section, the graded lexical entailment task has been proposed with a graded assertion [139]. In this section, we evaluate the proposed HWR and prior work on the graded lexical entailment task. The section is organised as follows. The description of the task is firstly presented in Section 5.4.2.1. The benchmark dataset and the results are then provided, respectively, in Section 5.4.2.2 and Section 5.4.2.3.

5.4.2.1 Task Description

The hypernym relation naturally underlines the lexical entailment relation [139]. For example, the hypernym pair (*owl*, *bird*) underlines that *owl* entails the existence of the *bird*. The supervised hypernym identification task described in the previous section (Section 5.4.1) simplifies the judgement of the lexical entailment (i.e. the hypernym relation existence) between words into a *binary* decision rather than a *gradual* decision. That is, given a word-pair (x, y) we were identifying whether (x *entails* y) or not, instead of (x *to a certain degree entails* y).

For the *graded* lexical entailment task, we evaluated the word representations on making *graded* assertions about the lexical entailment between words through human judgements. The task works in a similar fashion to the word similarity measurement task (Section 3.4.1.1). Here, human annotators are given a set of word-pairs and are asked to rate (on a specified scale) answering the question: *to what degree does x entail y*. Next, the entailment score between the word vector

Entailment Score Function	**Directionality**	**Source**
$D_1(\boldsymbol{x},\boldsymbol{y}) = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{\|\|\boldsymbol{x}\|\| \cdot \|\|\boldsymbol{y}\|\|}$	symmetric	-
$D_2(\boldsymbol{x},\boldsymbol{y}) = (1 - D_1(\boldsymbol{x},\boldsymbol{y})) + (\frac{\|\|\boldsymbol{x}\|\| - \|\|\boldsymbol{y}\|\|}{\|\|\boldsymbol{x}\|\| + \|\|\boldsymbol{y}\|\|})$	asymmetric	[140]
$D_3(\boldsymbol{x},\boldsymbol{y}) = D_1(\boldsymbol{x},\boldsymbol{y}) * \frac{\|\|\boldsymbol{y}\|\|}{\|\|\boldsymbol{x}\|\|}$	asymmetric	[111]
$D_4(\boldsymbol{x},\boldsymbol{y}) = -(1 + \alpha(\|\|\boldsymbol{x}\|\| - \|\|\boldsymbol{y}\|\|))*$ $(arcosh(1 + 2\frac{\|\|\boldsymbol{x}-\boldsymbol{y}\|\|^2}{(1-\|\|\boldsymbol{x}\|\|^2)(1-\|\|\boldsymbol{y}\|\|^2)}))$	asymmetric	[114]

Table 5.3: Different lexical entailment score functions. In each function, x represents the *hyponym* word and y represents the *hypernym*, and $||.||$ is the $\ell 2$ norm.

representations learnt by a particular method for the two words in each pair is computed and then compared against the average entailment ratings assigned by the human annotators. If the degree of correlation between human entailment ratings and the entailment scores computed using the learnt word representations is high, we can conclude that the word representations capture hierarchy as perceived by humans.

To compute the entailment score between the word vector representations, a symmetric distance function such as the *cosine* might not be appropriate because lexical entailment is asymmetric in general. Therefore, there is a need for an asymmetric distance function that takes into account both vector norm and direction to provide correct entailment scores between word pairs. For this purpose, several asymmetric functions have been proposed. For a comprehensive comparison, we use all of the previously proposed score functions in this experiment. Table 5.3 lists these score functions used to infer the lexical entailment between words.

Pair	**Rating**
(*crocodile*, *animal*)	10.00
(*olive*, *food*)	9.58
(*avenue*, *road*)	8.05
(*competence*, *ability*)	7.73
(*cement*, *stone*)	6.37
(*ball*, *game*)	5.25
(*intelligence*, *knowledge*)	4.48
(*throat*, *neck*)	3.33
(*lion*, *tiger*)	2.12
(*food*, *pie*)	1.53
(*jupiter*, *pluto*)	0.55
(*chain*, *bucket*)	0.00

Table 5.4: Examples of word-pairs from HyperLex benchmark dataset.

5.4.2.2 Benchmark Dataset

To evaluate the proposed HWR and prior work on the *graded* lexical entailment task, we select the gold standard dataset HyperLex [139] to test how well the word representations capture hierarchy. HyperLex focuses on the relation of *graded* lexical entailment at a continuous scale rather than simplifying the judgements into a binary decision. It consists of 2616 word pairs where each pair is manually annotated with a score on a scale of $[0, 10]$ indicating the strength of entailment.

5.4.2.3 Results

Following the standard protocol for evaluating using the HyperLex dataset, we measure the Spearman (ρ) correlation coefficient between gold standard ratings and the predicted scores. Table 5.5 shows the results of the Spearman correlation coefficients of HWR and the other word representation methods on the HyperLex dataset against the human ratings. We can see from Table 5.5 that HWR is able to encode the hierarchical structure in the learnt representations, reporting better or comparable results to all other methods using all the score functions, except for LEAR. It is worth noting that, HyperVec, LEAR and Poincaré use *pairwise* hypernym relations in a similar spirit to the structure of the benchmark datasets, whereas the proposed HWR uses the entire hierarchical path. For example, 59% of the word pairs in HyperLex have been seen by LEAR as explicit hypernym pairs during the retrofitting process.

Moreover, Table 5.5 shows that the first two categories of methods that were not specifically designed to encode hierarchical information report very poor per-

Method	**Score Function**			
	D_1	D_2	D_3	D_4
CBOW	0.10	0.04	0.05	0.06
Skipgram	0.08	0.05	0.00	0.09
GloVe	0.05	0.13	0.10	0.06
JointReps	0.07	0.07	0.04	0.04
HyperVec	0.17	0.47	0.51	0.04
LEAR	**0.44**	**0.63**	**0.63**	0.21
Poincaré	0.28	0.22	0.21	0.24
HWR	0.27	0.48	0.35	**0.26**

Table 5.5: Results (Spearman's ρ) of HWR and other word representations methods on the HyperLex dataset using different entailment score functions.

formance as compared to the hierarchical specific methods (the third category in the table), which justifies the use of the graded lexical entailment task for evaluating the hierarchical representations. However, it should be noted that the HyperLex dataset is specifically designed to consider lexical entailment, which might not precisely reflect the hierarchy order between words. For instance, we have observed that in the HyperLex dataset, the pair (*cat*, *animal*) is assigned a score of 10 indicating the strongest relations of lexical entailment, and the pair (*cat*, *mammal*) is given 8.5, whereas in WordNet *mammal* is the direct hypernym of *cat* but *animal* is the ninth in the hierarchical path.

5.4.3 Unsupervised Hypernym Detection and Directionality

To further evaluate the HWRs, we conduct another standard classification-style task. The unsupervised detection works in a similar fashion to the supervised experiment in Section 5.4.1.1 but unsupervisedly. In this experiment, we evaluate the word representations on *unsupervised* hypernym detection and directionality. To understand the task, Section 5.4.3.1 firstly provides a detailed description of it and how it differs from the supervised evaluation task presented earlier in Section 5.4.1. Next, Section 5.4.3.2, presents the benchmark datasets adopted in this task. The evaluation results (Section 5.4.3.3) then conclude the section.

5.4.3.1 Tasks Description

As noted above, the unsupervised detection task is similar to the supervised one but works in an unsupervised manner. The task here is to detect the hypernym relations (one class) from other types of relations. To this end, we perform binary classification on a dataset containing pairs of different semantic relations by randomly sampling 2% of the hypernymy pairs, and use this to learn a threshold by computing the average score, and then use the remaining 98% for testing. For computing the average score, we use all of the score functions given in Table 5.3. For the unsupervised directionality, the aim is to predict the hypernym word from each given hypernymy pair by comparing the vector norms of the words, where the larger norm indicates the hypernym and the smaller indicates the hyponym. Next, the prediction accuracy is reported as the performance measure.

5.4.3.2 Benchmark Datasets

For the unsupervised hypernym directionality, we follow the standard practice and select a subset of 1337 pairs extracted from the previously used (Section 5.4.1.2) Bless dataset. Similarly, for the unsupervised hypernym detection, WBless [142] dataset were selected which consists of a subset of 1668 pairs from Bless with different semantic relations including *hypernymy*, *meronymy* and *holonymy*.

Method	Bless	WBless			
		D_1	D_2	D_3	D_4
CBOW	21.03	47.96	42.15	44.18	36.45
Skipgram	23.61	47.18	45.44	43.65	37.47
GloVe	51.93	46.10	46.40	47.00	51.92
JointReps	34.12	47.24	44.84	45.56	47.90
HyperVec	94.02	52.4	59.95	**71.04**	**66.49**
Poincaré	40.68	55.14	50.12	54.32	49.88
LEAR	96.37	55.47	**70.44**	70.32	59.95
HWR	**97.52**	**55.62**	59.77	62.65	59.31

Table 5.6: Accuracy for unsupervised hypernym directionality (Bless) and detection (WBless). Different score functions are used in the detection task.

5.4.3.3 Results

Table 5.6 shows that HWR reports the best performance on the directionality task on the Bless benchmark dataset. We can also notice a large difference in the performance between the methods in the first two categories (non-hierarchical) of models as compared to the hierarchical methods. In particular, non-hierarchical models suffer when distinguishing between the two words in each pair, and assigning the narrower (hyponym) word a larger norm. In WBless, the experiment shows that HWR reports the best performance using D_1 and LEAR reports the best score on D_2, whereas by using D_3 and D_4, HyperVec achieves the best performance. Similar to the previous observation found in Section 5.4.2.3, it is noteworthy that since both HyperVec and LEAR use the pairwise hypernym relation constraints during the training, as such a large number of data might have already been seen explicitly as pairs. For instance, we have observed that 91% of the pairs in WBless are in the hypernym constraints given to LEAR during the retrofitting process.

5.4.4 Hierarchical Path Prediction

The supervised hypernym detection task presented in Section 5.4.1, the graded lexical entailment task in Section 5.4.2 and the unsupervised hypernym detection in Section 5.4.3, provide only a partial evaluation with respect to hierarchy. That is because all benchmark datasets used in those tasks are limited to *pairwise* datasets, and are annotated for hypernymy between two words, ignoring the *full* taxonomic structure. To address this issue, and inspired by the word analogy prediction task that is widely used to evaluate word representations (discussed in details in Section 3.4.2 in Chapter 3), we propose a *hierarchical path prediction* task. The task is introduced in this section which is organised as follows. Section 5.4.4.1 presents a description of the proposed task. Next, in Section 5.4.4.2, we introduce the proposed benchmark dataset that is used in this task. That is then followed by Section 5.4.4.3 where the results are presented and discussed. In Section 5.4.4.4, an ablation study is conducted using the proposed task and dataset. The section is then concluded with a qualitative analysis (Section 5.4.4.5).

5.4.4.1 Task Description

The hierarchical path prediction task aims to predict a hyponym word that fits best to complete a hierarchical path. That is, for a hierarchical path:

$$a \rightarrow b \rightarrow c \rightarrow d \rightarrow e$$

where b, c, d and e are hypernyms of a, the task is to predict the hyponym a given b, c, d and e. For example, revisiting a hierarchical path from Figure 5.1:

$$? \rightarrow \textit{vertebrate} \rightarrow \textit{chordate} \rightarrow \textit{animal} \rightarrow \textit{organism}$$

The task here is to predict the hyponym word that fits best to complete this hierarchical path, which is *bird* in this example. If there are multiple candidates (hyponyms a) with the same path, then we consider all such a's as correct answers to the hierarchical path prediction task. In fact, in the WordNet, there are on average 8 hyponym words ending with the same hierarchical path.

Two different methods can be used to predict a from a given path $b \rightarrow c \rightarrow d \rightarrow e$ as described next:

(i) *Compositional* (COMP) method predicts the word a from a given vocabulary that returns the highest score of $COMP(a, b \rightarrow c \rightarrow d \rightarrow e) = D_i(\boldsymbol{a}, \boldsymbol{b}) + D_i(\boldsymbol{a}, \boldsymbol{c}) + D_i(\boldsymbol{a}, \boldsymbol{d}) + D_i(\boldsymbol{a}, \boldsymbol{e})$.

(ii) Direct Hypernym (DH) method selects the word a that returns the highest score of $DH(a, b \rightarrow c \rightarrow d \rightarrow e) = D_i(\boldsymbol{a}, \boldsymbol{b})$ with only the vector of the direct hypernym b used to predict a.

Here $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, $\boldsymbol{d}$, and $\boldsymbol{e}$ indicate the vector representations of the corresponding words. For both COMP and DH, D_i can be any score function from Table 5.3.

5.4.4.2 Benchmark Datasets

We create a novel dataset by first sampling paths from WordNet which connects a hypernym to a hyponym. We limit the paths to contain words that are unigrams, bigrams or trigrams, and sample the paths including words with a broad range of frequencies. Moreover, no full path that is used as training data when computing $J_{\mathcal{T}}$ in Equation 5.1 is used when creating a dataset containing 330 paths. We further classify the paths in the dataset into unigram (containing only unigrams), bigram (contains at least one bigram but no trigrams), or trigram (containing at least one trigram) paths. There are respectively 150, 120 and 60 unigram, bigram and trigram paths in the created dataset.

5.4.4.3 Results

In Table 5.7, we report the accuracies (i.e. the percentages of the correctly predicted paths) for different word representation learning methods and prediction methods. According to the Clopper-Pearson confidence intervals [33] computed at $p < 0.05$, the proposed HWR method significantly outperforms all the other word representation learning methods compared in Table 5.7, irrespective of the prediction method or the score function being used. In contrast to the results in the previous tasks, where the prior word representations learning methods, including hierarchical methods such as HyperVec and LEAR, were performing constantly

Method	**Path Prediction Method**									
	COMP					DH				
	Score Function									
	D_1	D_2	D_3	D_4	${D_4}^*$	D_1	D_2	D_3	D_4	${D_4}^*$
CBOW	38.12	28.75	43.33	1.04	18.33	48.54	45.42	45.42	1.04	3.04
Skipgram	37.08	30.83	37.08	1.04	29.79	42.29	40.21	40.21	1.04	38.12
GloVe	28.75	21.46	27.71	0.0	19.38	46.46	40.21	41.25	1.04	40.21
JointReps	29.79	38.12	38.12	1.04	32.92	41.25	44.38	50.62	1.04	41.25
HyperVec	33.54	21.04	21.04	1.04	27.29	47.08	21.04	21.04	1.04	38.75
LEAR	67.29	19.38	22.5	**2.08**	16.25	78.75	22.5	22.5	0.0	3.04
Poincaré	75.3	65.61	59.85	0.0	48.33	76.21	63.18	60.76	0.0	68.03
HWR	**83.82**	**83.82**	**82.36**	0.30	**62.97**	**84.79**	**75.03**	**71.85**	0.61	**69.39**

Table 5.7: Accuracy (%) of the different word representations learning methods on the hierarchical path prediction dataset using the COMP and DH as prediction methods on different score functions over the hierarchical paths. The reported results are the average accuracy scores for the n-grams paths.

well on pairwise hypernymy datasets, they seem unable to encode the full hierarchical path. Moreover, Table 5.7 shows that Poincaré, which was not able to perform well in all previous tasks, performs much better in this task outperforming other methods, except HWR.

With COMP, HWR reports an average improvement of 16% in accuracy over Poincaré, which is the highest among the remaining methods. DH significantly improves the results for all word representations when using the scoring D_1 function. More importantly, the scoring functions D_2, D_3, and D_4 that have been proposed in prior work (Table 5.3) mainly for the graded lexical entailment task struggle to generalise to tasks that require inference with hierarchical word representations. For example, Table 5.7 shows that D_2 and D_3 perform significantly worse for all word representations models except for Poincaré and HWR. Further, it appears that some of such score functions are motivated by heuristic assumptions. In particular, in Table 5.7, we can see that applying D_4 performs remarkably poor for hierarchical path prediction, failing to correctly predict even a single path in most cases. Interestingly, dropping the $(1 + \alpha(||\boldsymbol{x}|| - ||\boldsymbol{y}||))$ term from D_4 and using only the hyperbolic distance (denoted by ${D_4}^*$) results in an improved performance as shown in Table 5.7.

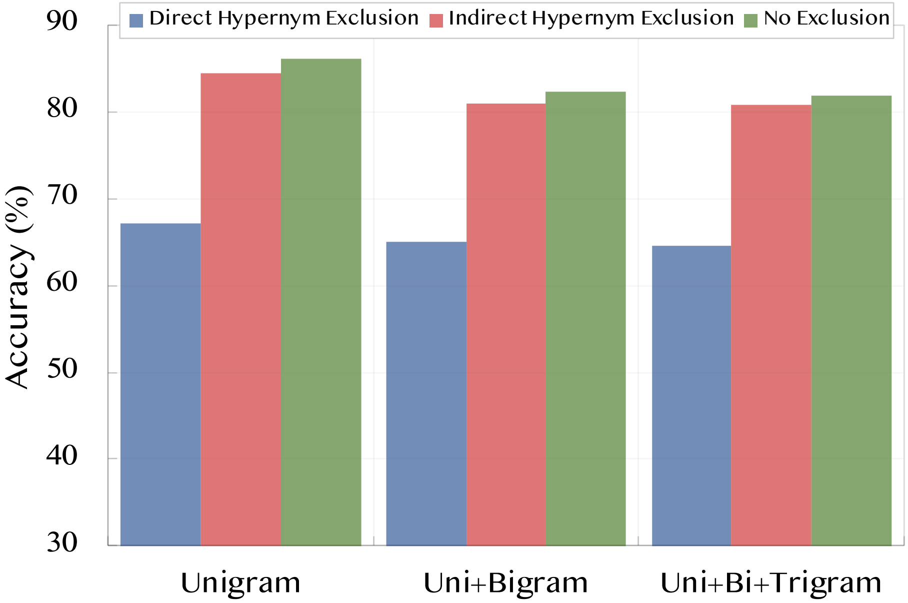


Figure 5.2: Impact of direct and indirect hypernym exclusion from a word's path evaluated on the hierarchical path prediction dataset with n-gram paths.

5.4.4.4 Ablation Study

To evaluate the effect of the direct hypernym b vs. indirect hypernyms (c, d, e) for predicting a, we conduct an ablation experiment using the COMP method on the hierarchical path prediction dataset over the different n-gram path categories. Specifically, given the path $a \rightarrow b \rightarrow c \rightarrow d \rightarrow e$, we use $D_i(\boldsymbol{a}, \boldsymbol{c}) + D_i(\boldsymbol{a}, \boldsymbol{d}) + D_i(\boldsymbol{a}, \boldsymbol{e})$ to compute $COMP(a, b \rightarrow c \rightarrow d \rightarrow e)$ for predicting a. Note that $D_i(\boldsymbol{a}, \boldsymbol{b})$ was excluded. We refer to this as the *Direct Hypernym Exclusion*. Whereas, removing exactly one out of $D_i(\boldsymbol{a}, \boldsymbol{c})$, $D_i(\boldsymbol{a}, \boldsymbol{d})$ and $D_i(\boldsymbol{a}, \boldsymbol{e})$ in the COMP method ($D_i(\boldsymbol{a}, \boldsymbol{b})$ is always used) is referred to as the *Indirect Hypernym Exclusion*. The COMP method that uses all the hypernyms $D_i(\boldsymbol{a}, \boldsymbol{b}) + D_i(\boldsymbol{a}, \boldsymbol{c}) + D_i(\boldsymbol{a}, \boldsymbol{d}) + D_i(\boldsymbol{a}, \boldsymbol{e})$ is shown as the *No Exclusion*. From Figure 5.2, we see that excluding the direct hypernym significantly decreases the accuracy of the prediction. This result supports our hypothesis that the direct hypernym carries vital information for the prediction of a hyponym in a path.

5.4.4.5 Qualitative Analysis

To demonstrate the ability of the proposed method for predicting the hierarchical paths, we qualitatively analyse the predictions of HWR and Poincaré, which report the best accuracy among all the other methods according to Table 5.7. A few randomly selected examples are shown in Table 5.8. The hyponym column represents gold standard answers (i.e. correct hyponym words). We show only a maximum of 5 correct hyponyms in Table 5.8. If a particular path has more than 5 hyponyms, we randomly select 5; otherwise, all possible hyponyms are listed.

From Table 5.8, we see that HWR accurately predicts the correct word in many cases where Poincaré fails (shaded rows in the table). Moreover, Poincaré in different cases tends to predict closely related words, but not precisely completing the hierarchical path. For examples, given the path (? → *head_ dress* → *clothing* → *consumer_ goods* → *commodity*), HWR correctly predicts the missing word to be *hat*, whereas Poincaré incorrectly predicts *muff*, which is for *hands* rather than *head*. Similarly, Table 5.8 shows that HWR is able to consider all the hypernym words when predicting the hyponym, but Poincaré was incapable. For example, for the path (? → *financial_ condition* → *condition* → *state* → *attribute*) HWR correctly predicts the hyponym word *wealth*, whereas Poincaré wrongly predicts *enjoyment* which is a *state* and an *attribute* but not a *financial_ condition*. Further, HWR shows an ability to accurately preserve the hierarchical order in the path whereas Poincaré fails. For instance, HWR was able to predict *feline* to complete the path (? → *carnivore* → *placental* → *mammal* → *vertebrate*) but Poincaré predicts *jaguar*, which is in fact a *carnivore* but in a lower order to *feline* as recorded in WordNet.

Moreover, Table 5.8 shows some cases where both HWR and Poincaré correctly predict the hyponym candidate. For example, given the path (? → *religious* → *religious_ person* → *person* → *casual_ agent*), both HWR and Poincaré managed to predict the correct hyponym word *monk*. Furthermore, from Table 5.8, we can also see that in some cases, HWR struggled to predict the correct words, while Poincaré has managed to complete the path accurately. For example, HWR failed to predict the word(s) *temple*, *mosque*, *bethel*, *masjid* or *chapel* to complete the path (? → *place_ of_ worship* → *building* → *structure* → *artifact*) while Poincaré was able to do so.

Hypernym$_1$ (b)	**Hypernym$_2$ (c)**	**Hypernym$_3$ (d)**	**Hypernym$_4$ (e)**	**Hyponym(s) (a's)**	**HWR prediction**	**Poincaré prediciton**
container	instrumentality	artifact	whole	scuttle, dispenser, dish, basket, capsule	dish	car
headdress	clothing	consumer_goods	commodity	cap, kaffiyeh, hat, topknot, turban	hat	muff
carnivore	placental	mammal	vertebrate	feline,viverrine procyonid	feline	jaguar
opinion	belief	content	cognition	judgment, eyes, preconception	judgment	waiting_game
physical_property	property	attribute	abstraction	luminosity, randomness, weight, invisibility, perceptibility	weight	apathy
financial_condition	condition	state	attribute	wealth, poverty, credit_crunch, solvency, tight_money	wealth	enjoyment
path	line	location	object	beeline, direction, traffic_pattern, migration_route, trail	direction	reservation
philosophy	humanistic_discipline	discipline	knowledge_domain	axiology, dialectic, logic, metaphysics, epistemology	logic	physics
paper	material	substance	matter	confetti, wax_paper, oilpaper, card, wallpaper	card	pigment
concession	contract	written_agreement	agreement	franchise	franchise	premise
air_defense	defense	military_action	group_action	active_air_defense, passive_air_defense	active_air_defence	war
food	solid	matter	physical_enti	junk_food, seafood, fresh_food, leftovers,meat	sea_food	dish
bicycle	wheeled_vehicle	container	instrumentali	safety_bicycle, velocipede, mountain_bike	mountain_bike	car
constructive_fraud	fraud	crime	transgression	fraud_in_law	fraud_in_law	fraud_in_law
religious	religious_person	person	causal_agent	monk, friar, eremite,votary, nun	monk	monk
footwear	covering	artifact	whole	slipper, flats, shoe, clog, boot, overshoe	boot	boot
place_of_worship	building	structure	artifact	masjid, mosque, temple, bethel, chapel	theatre	mosque

Table 5.8: Selected predictions of HWR and Poincaré on the hierarchical path prediction task (COMP). Hyponym(s) represents gold standard answer(s).

w	α	x	β	y	γ	z
pizza	0.12	cheese	0.65	flour	0.17	tomato
pizza	0.25	cheese	0.74	flour	0.00	sugar
biryani	0.73	chili	0.05	chicken	0.22	rice
biryani	0.00	sugar	0.24	chicken	0.64	rice
sushi	0.26	butter	0.00	avocado	0.68	salmon
sushi	0.18	butter	0.21	rice	0.61	salmon
coffee	0.52	liquid	0.20	beans	0.17	sodium
coffee	0.76	liquid	0.23	beans	0.00	protein
king	0.16	royal	0.84	man	0.00	woman
queen	0.25	royal	0.00	man	0.75	woman
king	0.11	crown	0.89	man	0.00	woman
queen	0.08	crown	0.00	man	0.92	woman

Table 5.9: Examples of decomposed hierarchical word representations.

5.4.5 Word Decomposition

To further evaluate the proposed HWR, we would like to understand how the meaning of a word can be related to the meanings of its parent concepts. For this purpose, we propose an evaluation method that expresses the hierarchical word representations of a word as the linearly-weighted combination over a set of given words. Specifically, given a word w and three anchor words x, y, z, we find their weights respectively α, β and γ such that the squared $\ell 2$ loss given by Equation 5.6 is minimised. Note that, unlike in the hierarchical path completion task, here we do not require x, y, z to be on the same hierarchical path as w.

$$L(\alpha, \beta, \gamma; \boldsymbol{w}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = ||\boldsymbol{w} - \alpha\boldsymbol{x} - \beta\boldsymbol{y} - \gamma\boldsymbol{z}||_2^2 \tag{5.6}$$

Minimisers of α, β and γ are found via Stochastic Gradient Descent and are subsequently normalised to unit sum.

Some example decompositions are shown in Table 5.9. For example, we see that *pizza* has *cheese*, *flour* and *tomato* components but not *sugar*. Similarly, *sushi* has *butter*, *rice* and *salmon* but not *avocado*. We can also see that both *king* and *queen* have a *crown* and *royal* components but the former has a *man* component while the latter has a *woman* component.

5.5 Summary

In this chapter, a *Hierarchical Word Representation* (HWR) method for fine-tuning the word representations to encode hierarchical information using a KB and a corpus was proposed. Unlike prior work that learned hierarchical word representations focusing on pairwise hypernym relations, the proposed HWR utilise the full hierarchical path of words from the taxonomy. To incorporate the corpus information into the learning process, the proposed HWR uses a hybrid of count- and prediction-based objective, instead of purely relying on prediction-based approach.

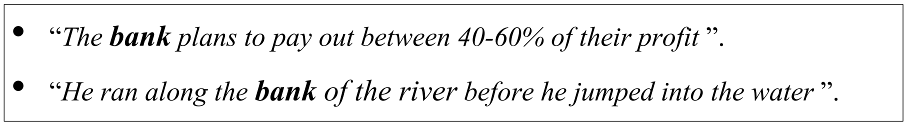
The proposed method HWR was evaluated on several standard tasks such as supervised and unsupervised hypernym detection and graded lexical entailment tasks on several benchmark datasets. Alongside the standard tasks, a novel task that is explicitly designed to evaluate the full hierarchical structure between words was also proposed. In this task, we evaluate how well word representations methods do to accurately predict hyponyms that complete hierarchical paths in a taxonomy. Moreover, a word decomposition task that attempts to evaluate the word representations on understanding how the meaning of a word can be related to the meanings of its parent concepts was also proposed. In all tasks, the proposed method HWR shows the ability to encode hierarchical information into the learnt word representations. In addition, the two newly-proposed evaluation tasks reveal that the current standard tasks that are used to evaluate the hierarchical relation between word might not be sufficient as they mainly focus on pairwise relations (lexical entailment between two words) rather than the full hierarchical path.

In the following chapter, the third proposed joint approach, SAWR, concerning learning sense-aware word representations, is presented. In SAWR, an unlabelled corpus and a sense-labelled corpus, in which the senses are linked with a KB, are used to learn the sense representations.

Chapter 6

Sense-Aware Word Representation

6.1 Introduction



- "*The* ***bank*** *plans to pay out between 40-60% of their profit* ".
- "*He ran along the* ***bank*** *of the river before he jumped into the water* ".

The lexical ambiguity (the presence of a word having multiple meanings) is one of the key challenges in natural language understanding. Let us revisit the example provided earlier in Chapter 1 (Section 1.2): The word *bank* in the first sentence refers to the *financial institution*, while the *bank* in the second sentence refers to the *river-bank.* Humans can easily identify the meaning of such ambiguous words by looking into the context they are used in, but computers might find it difficult to distinguish between such difference in the meaning, hence identifying the implied meaning of ambiguous words plays a fundamental rule in NLP. Although the typical corpus-based word representation learning approaches and the *joint* approaches such as the two so far purposed in this thesis (JointReps and HWR) have shown significant effectiveness on capturing the meaning of words in many NLP tasks, they are still struggling with lexical ambiguity. Specifically, a common limitation associated with existing methods is that they represent each word by a *single* vector, ignoring the potentially multiple senses of a word (polysemy or homonymy). Revisiting our example above, the two senses of *bank* are significantly different, and learning a single vector representation for both senses is inadequate. Consequently, several solutions have been proposed in the literature to overcome this limitation and learn *sense representations*, which capture the sense related information of words.

In this chapter, we intend to address the polysemy problem and present the third joint approach proposed in this thesis. Specifically, we propose a *Sense-Aware Word Representations* (SAWR) approach that jointly learns from both unlabelled and sense-tagged (where the senses are tagged using a KB) corpora. The proposed method, SAWR, can learn both word and sense representations in the same vector space by efficiently exploiting both types of resources.

As indicated above, learning sense-aware word representations have attracted the attention of several research studies (discussed in details in Section 2.3.3.3, Chapter 2). However, the joint approaches proposed previously for learning sense-aware word representations either directly using a KB (sense inventories defining the different senses of a word); or using word-sense taggers, that can be applied to unlabelled corpora, to generate automatic sense-labelled training data.

To the best knowledge of the author, none of the prior work on learning sense-aware word representations has attempted to use manually sense-tagged corpora, because such resources are either underdeveloped or not available on a large scale [73]. However, considering the fact that methods that learn only word representations such as CBOW, Skipgram and GloVe etc. can operate on unlabelled corpora, it remains unclear whether unlabelled data can help the process of learning sense representations, thereby reducing the manual effort required for creating sense tagged corpora for learning sense representations. Revisiting our previous example, only few instances of the word *bank* might be annotated in the labelled data with its sense as a *financial institute*, however, there might be many other words such as *cash*, *ATM*, *transaction* etc. that co-occur with *bank* that could contribute useful information about this particular sense towards the representation of *bank*. Importantly, such word-level co-occurrences can be obtained purely using unlabelled texts, which are comparatively easier to obtain than sense-labelled texts.

Hence, in this chapter's proposed method, SAWR, we use both a large unlabelled corpus and a comparatively smaller manually sense-labelled corpus to learn both word and sense representations simultaneously. It is worth mentioning that the standard practice for creating a manually sense-labelled corpus is that each word is linked to its appropriate sense in a KB and that allows the proposed SAWR to practically benefit from both a corpus and a KB. The chapter commences, Section 6.2, with the learning process of SAWR and then goes on to present the experimental setup and training details (Section 6.3). Next, Section 6.4 provides an extensive evaluation of the proposed method in several tasks. The chapter is concluded with a summary in Section 6.5.

6.2 Learning Process of SAWR

We propose SAWR, a method that jointly learns word and sense representations in the same lower-dimensional dense vector space. To explain the proposed SAWR, let us consider the lemma of the *target word* $w_i \in \mathcal{V}$ for which we are interested in learning a word representation $\boldsymbol{w}_i \in \mathbb{R}^d$ in some d-dimensional real space. Here, $\mathcal{V}$ is the vocabulary of words and we use bold fonts to denote word/sense representation vectors. Given an unlabelled (i.e. not sense-tagged) corpus $\mathcal{C}$, let us denote the set of contexts in which w_i occurs by $\mathcal{K}_i$. Here, for example, a context can be a window of fixed/dynamic length, a sentence or a document. Next, let us consider the lemma of the *context word* w_j that co-occurs with w_i, denoted by $w_j \in \mathcal{K}_i$. Inspired by the negative sampling method used in Skipgram, we would like to learn the representations of w_i and w_j close to each other than a word $w_m (\notin \mathcal{K}_i)$ that does not co-occur with w_i. We sample $w_m \sim P_u$ from the unigram distribution P_u such that words that are frequent in the corpus (therefore likely to occur in a given sentence) but do not co-occur with w_i as the *negative* samples. We define the hinge loss J_{ww} for predicting w_j over w_m in all contexts $\mathcal{K}(w_i)$ over the entire vocabulary by:

$$J_{ww} = \sum_{w_i \in \mathcal{V}} \sum_{w_j \in \mathcal{K}_i} \sum_{\substack{w_m \sim P_u \\ w_m \notin \mathcal{K}_i}} \max\left(-{\boldsymbol{w}_i}^\top \boldsymbol{w}_j + {\boldsymbol{w}_i}^\top \boldsymbol{w}_m + 1, 0\right) \tag{6.1}$$

J_{ww} can be computed using unlabelled data and does not involve sense representations.

We require that the word representations must be able to predict not only the co-occurrences of a context word in contexts where a target word occurs, but also must be able to predict the senses associated with the target and contexts words. To model such word vs. sense co-occurrences, given a sense-tagged corpus $\mathcal{G}$, we compute the hinge loss J_{ws} associated with predicting the correct sense s_{jt} of the context word w_j and a randomly sampled sense s_{mg} from the distribution of senses in unigrams P_s that does not occur with w_i as follows:

$$J_{ws} = \sum_{w_i \in \mathcal{V}} \sum_{s_{jt} \in \mathcal{K}_i} \sum_{\substack{s_{mg} \sim P_s \\ s_{mg} \notin \mathcal{K}_i}} \max \left(-\boldsymbol{w_i}^\top \boldsymbol{s}_{jt} + \boldsymbol{w}_i{}^\top \boldsymbol{s}_{mg} + 1, 0 \right) \tag{6.2}$$

Here, P_s is computed by counting the occurrences of senses in $\mathcal{G}$.

Likewise, we can compute the hinge loss J_{sw} for predicting a context word using the sense s_{iy} of the target word w_i over a randomly sampled word $w_m \sim P_u$ as follows:

$$J_{sw} = \sum_{w_i \in \mathcal{V}} \sum_{w_j \in \mathcal{K}_i} \sum_{\substack{w_m \sim P_u \\ w_m \notin \mathcal{K}_i}} \max \left(-\boldsymbol{s}_{iy}{}^\top \boldsymbol{w}_j + \boldsymbol{s}_{iy}{}^\top \boldsymbol{w}_m + 1, 0 \right) \tag{6.3}$$

Finally, we require that sense representations must be able to predict the correct sense s_{jt} of a context word w_j given the sense s_{iy} of the target word w_i. This requirement is captured by the hinge loss given by Equation 6.4, where the inner-product between $\boldsymbol{s}_{iy}$ and $\boldsymbol{s}_{jt}$ must be greater than with $\boldsymbol{s}_{mg}$, a randomly sampled sense $s_{mg} \sim P_s$, as given by Equation 6.4.

$$J_{ss} = \sum_{l_i \in \mathcal{V}} \sum_{s_{nt} \in \mathcal{K}_i} \sum_{\substack{s_{mg} \sim P_s \\ s_{mg} \notin \mathcal{K}_i}} \max \left(-\boldsymbol{s}_{iy}{}^\top \boldsymbol{s}_{jt} + \boldsymbol{s}_{iy}{}^\top \boldsymbol{s}_{mg} + 1, 0 \right) \tag{6.4}$$

We combine the four objective functions given above into a single linearly-weighted objective given by Equation 6.5, for some $\lambda_1, \lambda_2, \lambda_3 \in \mathbb{R}$ coefficients:

$$J = J_{ww} + \lambda_1 J_{ws} + \lambda_2 J_{sw} + \lambda_3 J_{ss} \tag{6.5}$$

We find the word representations $\boldsymbol{w}_i$, $\boldsymbol{w}_j$, $\boldsymbol{w}_m$ and sense representations $\boldsymbol{s}_{iy}$, $\boldsymbol{s}_{jt}$, $\boldsymbol{s}_{mg}$ such that J is minimised. For this purpose, we compute the partial derivatives of J with respect to word and sense representations. The derivatives of the objective function given by Equation 6.5 with respect to the variables are given as follows:

$$\frac{\partial J}{\partial \boldsymbol{w}_i} = \begin{cases} \begin{cases} -\boldsymbol{w}_j + \boldsymbol{w}_m & \text{if} \quad \boldsymbol{w}_i^\top(\boldsymbol{w}_j - \boldsymbol{w}_m) \leq 1 \\ 0 & \text{otherwise} \end{cases} \\ +\lambda_1 \begin{cases} -\boldsymbol{s}_{jt} + \boldsymbol{s}_{gm} & \text{if} \quad \boldsymbol{w}_i^\top(\boldsymbol{s}_{jt} - \boldsymbol{s}_{gm}) \leq 1 \\ 0 & \text{otherwise} \end{cases} \end{cases} \tag{6.6}$$

$$\frac{\partial J}{\partial \boldsymbol{w}_j} = \begin{cases} \begin{cases} -\boldsymbol{w}_i & \text{if} \quad \boldsymbol{w}_i^\top(\boldsymbol{w}_j - \boldsymbol{w}_m) \leq 1 \\ 0 & \text{otherwise} \end{cases} \\ +\lambda_2 \begin{cases} -\boldsymbol{s}_{iy} & \text{if} \quad \boldsymbol{s}_{iy}^\top(\boldsymbol{w}_j - \boldsymbol{w}_m) \leq 1 \\ 0 & \text{otherwise} \end{cases} \end{cases} \tag{6.7}$$

$$\frac{\partial J}{\partial \boldsymbol{w}_m} = \begin{cases} \begin{cases} \boldsymbol{w}_i & \text{if} \quad \boldsymbol{w}_i^\top(\boldsymbol{w}_j - \boldsymbol{w}_m) \leq 1 \\ 0 & \text{otherwise} \end{cases} \\ +\lambda_2 \begin{cases} \boldsymbol{s}_{iy} & \text{if} \quad \boldsymbol{s}_{iy}^\top(\boldsymbol{w}_j - \boldsymbol{w}_m) \leq 1 \\ 0 & \text{otherwise} \end{cases} \end{cases} \tag{6.8}$$

$$\frac{\partial J}{\partial \boldsymbol{s}_{iy}} = \begin{cases} \lambda_2 \begin{cases} -\boldsymbol{w}_j + \boldsymbol{w}_m & \text{if} \quad \boldsymbol{s}_{iy}^\top(\boldsymbol{w}_j - \boldsymbol{w}_m) \leq 1 \\ 0 & \text{otherwise} \end{cases} \\ +\lambda_3 \begin{cases} -\boldsymbol{s}_{jt} + \boldsymbol{s}_{gm} & \text{if} \quad \boldsymbol{s}_{iy}^\top(\boldsymbol{s}_{jt} - \boldsymbol{s}_{gm}) \leq 1 \\ 0 & \text{otherwise} \end{cases} \end{cases} \tag{6.9}$$

$$\frac{\partial J}{\partial \boldsymbol{s}_{jt}} = \begin{cases} \lambda_1 \begin{cases} -\boldsymbol{w}_i & \text{if} \quad \boldsymbol{w}_i^\top(\boldsymbol{s}_{jt} - \boldsymbol{s}_{gm}) \leq 1 \\ 0 & \text{otherwise} \end{cases} \\ +\lambda_3 \begin{cases} -\boldsymbol{s}_{iy} & \text{if} \quad \boldsymbol{s}_{iy}^\top(\boldsymbol{s}_{jt} - \boldsymbol{s}_{gm}) \leq 1 \\ 0 & \text{otherwise} \end{cases} \end{cases} \tag{6.10}$$

$$\frac{\partial J}{\partial \boldsymbol{s}_{gm}} = \begin{cases} \lambda_1 \begin{cases} \boldsymbol{w}_i & \text{if} \quad \boldsymbol{w}_i^\top(\boldsymbol{s}_{jt} - \boldsymbol{s}_{gm}) \leq 1 \\ 0 & \text{otherwise} \end{cases} \\ +\lambda_3 \begin{cases} \boldsymbol{s}_{iy} & \text{if} \quad \boldsymbol{s}_{iy}^\top(\boldsymbol{s}_{jt} - \boldsymbol{s}_{gm}) \leq 1 \\ 0 & \text{otherwise} \end{cases} \end{cases} \tag{6.11}$$

6.3 Experiments

The experimental settings that have been used to train the proposed SAWR method are presented in this section. The section commences with details of the data that was used to train the model (Section 6.3.1). The following section (Section 6.3.2) then provide the model setup and the training process with respect to the model's hyperparameters and the optimisation.

6.3.1 Training Data

To train the proposed method, SAWR, we use both a large unlabelled corpus and a comparatively smaller manually sense-labelled corpus. As the unlabelled corpus in our experiments, we used the ukWaC which was discussed in detail in Section 3.3.1. For the manually sense-labelled corpus, we used SemCor [105]. SemCor is a subset of the English Brown corpus [81] with approximately $700,000$ tokens in which around $200,000$ of them are sense and POS tagged with reference to the WordNet, making it the largest available manually sense-labelled corpus.

Before training the proposed method with SemCor as a manually sense-labelled corpus, we wanted to verify that the proposed method can learn sense representations for the different senses of an ambiguous word as expected. For that purpose, we create a *pseudo sense-labelled* corpus by replacing either two or four words in the ukWaC by a unique identifier to create an *artificially* sense-labelled corpus. Details of this pseudo sense-labelled experiment are discussed later in this chapter (Section 6.4.1).

6.3.2 Model Setup and Training

The proposed SAWR randomly initialises each word w_i and each of its senses s_{iy} with unique representation vectors, and update those vectors such that the rank loss between words and senses that co-occur in unlabelled or labelled contexts is minimised over the entire vocabulary of words. The proposed method works in an *online* fashion, where we require only a single pass over the data considering one sentence at a time. We set the context window to 10 tokens to the right and left of a word in the sentence. We used 5 negative samples for both words $\boldsymbol{w}_m$ and senses $\boldsymbol{s}_{mg}$ with 0.75 as a uniform sampling rate. We use SGD as the optimisation method. The proposed model converged to a solution with 20 training epochs taking around 10 hours to learn 300 dimensional word representations for $|\mathcal{V}| = 99,663$ words on a Xeon 2.9GHz 32 core 512GB RAM machine.

To tune the hyperparameters λ_1, λ_2 and λ_3 given in Equation 6.5, we follow a similar procedure to the previously used in Section 3.4.1.3. Specifically, we used the Rubenstein-Goodenough (RG) [127] word similarity benchmark dataset as a validation dataset, in which we vary the values of the coefficients λ_1, λ_2 and λ_3 and learn the sense and word representations using the proposed method before measuring the Spearman correlation on RG dataset. Next, λ_1, λ_2 and λ_3 values are selected based on the highest reported correlation score. Setting $\lambda_1 = \lambda_2 = \lambda_3 = 10$ performed consistently well in our experiments.

6.4 Evaluation and Results

We conduct three sets of experiments to evaluate the sense-aware word representations learnt by the proposed SAWR. First, in Section 6.4.1, we qualitatively evaluate the ability of the proposed method to discover known senses in a pseudo-labelled dataset. Second, in Section 6.4.3, we compare the sense-aware word representations learnt by SAWR against prior work directed at the word similarity measurement task using an in-context benchmark dataset. Third, in Section 6.4.4, we use the sense-aware word representations learnt by SAWR to solve sentiment analysis task and compare the performance against prior work.

6.4.1 Qualitative Analysis

To firstly verify that the proposed method can learn sense representations for the different senses of an ambiguous word as expected, we conducted the following experiment. We create a *pseudo* sense-tagged corpus by replacing all occurrences of *two* words by an *artificial* word in a corpus and tag the mentions of original words as different senses of the artificial word. A few selected examples are shown in Table 6.1, where we select words with different frequencies (ratio of frequencies indicated within brackets in the first column). For example, we replace *career* and *africa* with the artificial ambiguous word *careeryafrica* with two senses (sense$_1$ and sense$_2$) corresponding *career* and *africa*. Using ukWaC as the unlabelled corpus, we produced a pseudo-labelled corpus following this procedure. This approach enables us to create arbitrarily large sense-tagged corpora with known senses (and frequencies), which is useful for verifying that the proposed method is working as expected.

We run the proposed method independently on the (a) unlabelled corpus and (b) the combination of unlabelled and pseudo-labelled corpora. Next, we compute the word (in the case of both (a) and (b)) and sense (in the case of (b) only) representations. The nearest neighbouring words (computed using the cosine similarity between the learnt 300-dimensional representations) for setting (a) (third column) and for setting (b) (fourth and fifth columns) are shown in Table 6.1.

From Table 6.1 we see that the nearest neighbours of the word representations learnt using only the unlabelled corpus are a mixture of the multiple senses of the ambiguous artificial word. For example, the nearest neighbours of the artificial word *careeryafrica* when using only the unlabelled corpus are a mixture of the neighbours of *career* and *africa* such as *south*, *australia*, *china*, *job* and *academic*. On the other hand, the sense representations learnt by the proposed method using both unlabelled and labelled data enable us to produce coherent neighbourhoods, capturing a single sense of the artificial ambiguous word.

Naturally, ambiguous words tend to have more than two different senses; hence, it is important for the proposed SAWR to be able to capture those multiple senses. For this purpose, in Table 6.2, we conduct the same experiment reported in Table 6.1 and described above, however with the difference that instead of replacing all occurrences of *two* words by an artificial word in a corpus, we replace all occurrences of *four* words. This approach allows us to verify the ability of the proposed method to capture the multiple senses of the ambiguous word. From Table 6.2 we can see that the proposed method was able to detect the correct single sense for each word (fourth, fifth, sixth and seventh column) using both labelled and unlabelled corpora, unlike the mixture of the various senses (third column) produced by using only unlabelled corpus. It is worth noting that even with the frequency variation (ratio of frequencies indicated within brackets in the first column) of the selected *four* words to be replaced by the artificial ambiguous word, the proposed method was able to capture the correct senses correctly.

Words	**Unique Identifier (ambigious word)**	**Nearest Neighbours unlabelled corpus**	**Nearest Neighbours joint corpora**	
			sense$_1$	**sense$_2$**
career (0.8) africa (0.2)	careeryafrica	south	careers	india
		australia	professional	europe
		development	profession	asia
		education	graduate	south
		professional	academic	kenya
		developing	employment	australia
		china	training	china
		west	development	african
		seeking	job	southern
		experience	successful	countries
		job	skills	pacific
		academic	pursue	brazil
stock (0.7) dance (0.3)	dancystock	market	stocks	dancing
		music	market	music
		shares	markets	musical
		exchange	price	jazz
		company	exchange	theatre
		rolling	purchase	art
		markets	prices	singing
		art	investment	ballet
		mix	company	drama
		stocks	shares	artists
		theatre	trading	opera
		dancing	products	song
sea (0.6) chapter (0.4)	chapterysea	river	ocean	introduction
		ocean	river	notes
		introduction	coast	chapters
		atlantic	mountains	book
		island	bay	summary
		coastal	atlantic	article
		section	shore	describes
		shore	beach	act
		coast	coastal	review
		above	island	section
		waters	sand water	paragraph
		north		report
dog (0.5) chairman (0.5)	dogychairman	executive	cat	executive
		cat	puppy	chief
		chief	pet	committee
		president	horse	director
		bob	cats	treasurer
		david	dogs	secretary
		director	rat	john
		john	girl	vice
		horse	breed	deputy
		cats	sheep	officer
		brown	horses	turner
		fox	boy	superintendent

Table 6.1: Nearest Neighbours of the sense (*two senses*) and word representations learnt by SAWR using a unlabelled and joint (labelled+unlabelled) corpora.

Words	Unique Identifier (ambigious word)	Nearest Neighbours unlabelled corpus	Nearest Neighbours joint corpora			
			sense$_1$	**sense$_2$**	**sense$_3$**	**sense$_4$**
national (0.5) road (0.2) forest (0.2) faith (0.1)	nationalroad	forests	international	lane	forests	belief
		woods	regional	junction	woods	christian
		regional	local	roads	valley	religion
		local	scottish	hill	forestry	god
		international	wales	street	park	christ
		scottish	british	park	deer	jesus
		lane	association	avenue	hills	christians
		junction	institute	bridge	habitat	religious
		roads	government	traffic	wood	christianity
		belief	agency	highway	trees	islam
		british	european	crossing	hill	gospel
		wales	scotland	mile	river	holy
policy (0.4) farm (0.2) family (0.2) medical (0.2)	policyfarm	policies	policies	farms	families	medicine
		farms	strategy	farmer	friends	dental
		farmer	government	farming	home	health
		families	issues	farmers	parents	healthcare
		medicine	strategic	dairy	relatives	physicians
		strategy	development	cattle	mother	doctors
		government	research	mill	father	clinical
		issues	framework	barn	children	nursing
		farmers	economic	sheep	house	physician
		friends	governance	cottage	husband	veterinary
		parents	public	village	wife	care
		dental	legislation	agricultural	child	specialist
minister (0.3) transport (0.3) blood (0.2) climate (0.2)	ministertrans	warming	prime	transportation	liver	warming
		prime	ministers	infrastructure	bleeding	emissions
		transportation	deputy	local	glucose	global
		deputy	secretary	bus	oxygen	pollution
		secretary	government	freight	kidney	environmental
		rail	mp	buses	fluid	ozone
		change	mr	passenger	skin	change
		liver	ministry	airports	cholesterol	greenhouse
		ministers	blair	public	lung	environment
		bleeding	spokesman	roads	tissue	carbon
		global	president	travel	stomach	weather
		freight	chancellor	rail	cells	impacts
council (0.3) album (0.3) floor (0.3) urban (0.1)	councilalbum	albums	borough	albums	floors	rural
		borough	councils	songs	basement	cities
		floors	county	song	roof	areas
		songs	district	band	bathroom	landscape
		councils	authority	ep	room	city
		basement	committee	debut	ceiling	sustainable
		song	housing	tunes	ground	development
		rural	community	cd	kitchen	spaces
		band	government	punk	bedroom	communities
		county	executive	singer	flat	towns
		debut	local	pop	flooring	transport
		district	authorities	tracks	lounge	suburban

Table 6.2: Nearest Neighbours of the sense (*four senses*) and word representations learnt by SAWR using a unlabelled and joint (labelled+unlabelled) corpora.

6.4.2 Visualisation

To illustrate the ability of the proposed method for learning the sense and word representations, we use t-SNE [95] to project the word representations learnt by SAWR using the pseudo sense-labelled corpus to a two-dimensional space as shown in Figure 6.1. Nearest neighbours of *dogychairman* and its two senses are highlighted. We see that the proposed method successfully learns the different senses of the ambiguous word in the embedding space. For example, the *dog* sense of *dogychaieman* has neighbours such as *dogs, cats* and *pet*, whereas the *chairman* sense has *executive, president* and *director* as the neighbours.

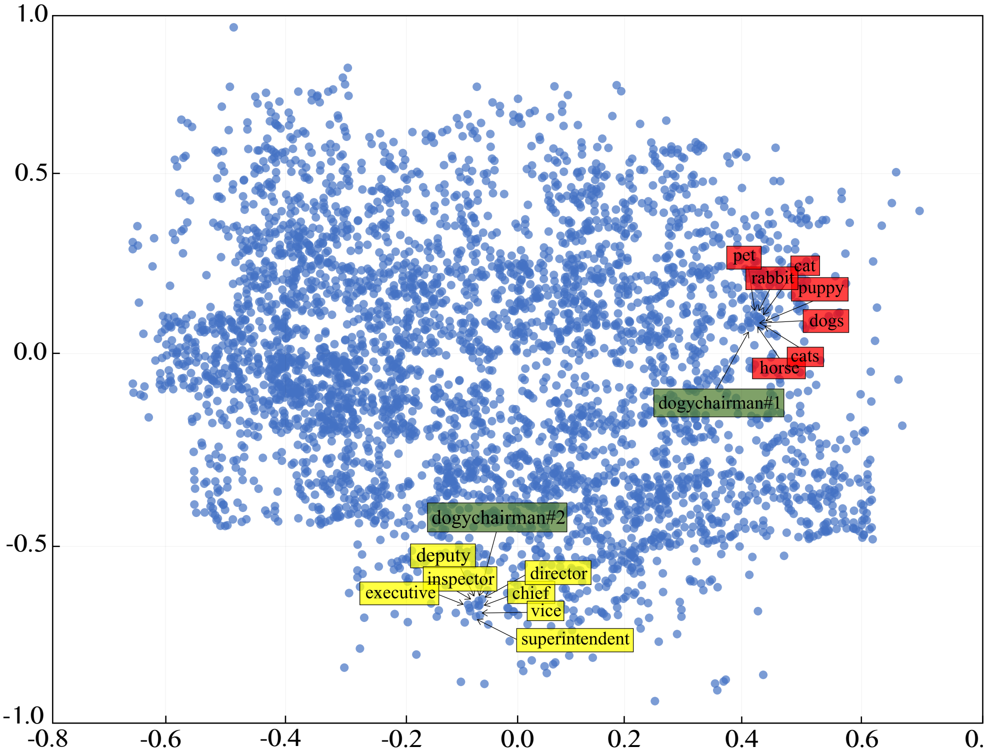


Figure 6.1: t-SNE projection in two-dimensional space of the word and sense representations learnt by the SAWR approach. Green labels show the two sense representations for artificial word *dogychairman*, whereas yellow and red labels show the nearest neighbours for the two senses.

6.4.3 Word Similarity

To further empirically evaluate the proposed SAWR and compare it against prior work, we used, following the same settings, the word similarity measurement task that was previously used to evaluate the first joint approach proposed in this thesis (described in details Section 3.4.1, Chapter 3). In which, we measure the cosine similarity between two words in human similarity benchmark datasets using their representations learnt by a word representation method, and then measure the Spearman correlation coefficient between human similarity ratings and computed cosine similarities. A higher correlation with human similarity ratings implies that the word representations learnt by the proposed method accurately capture the semantics of the words. For this task, out of the word similarity benchmark datasets discussed and used in Section 3.4.1.2 (Chapter 3), we adopted the SCWS benchmark dataset to evaluate SAWR, because it is the only benchmark dataset where the context (a full sentence) of each word-pair is provided, helping to point out the intended sense.

Method	**Spearman's ρ**
CBOW	0.523
Skipgram	0.582
GloVe	0.483
JointReps	0.510
Retro(Skipgram)	0.481
MSSG	0.632
NP-MSSG	**0.639**
SenseRetro	0.417
SAWR	0.606

Table 6.3: Performance of the proposed method, SAWR, in comparison with prior work evaluated on SCWS word similarity benchmark datasets.

In Table 6.3, we compare the performance of the proposed SAWR method against prior work such as: (i) sense-insensitive corpus-based word representations methods CBOW, Skipgram and GloVe, (ii) sense-insensitive joint approaches JointReps (the first joint approach proposed in this thesis (Chapter 3)) and Retro where we retrofit the vectors learnt by Skipgram (Section 2.3.3.1) and (iii) sense-sensitive representations MSSG, NP-MSSG and SenseRetro. Details of the prior sense-sensitive work were discussed in details in Section 2.3.3.3 (Chapter 2). For MSSG and NP-MSSG we used the publicly available source code and trained the methods on the same datasets as the proposed method. Unfortunately, the implementations nor trained word representations were available for SenseRetro method. Therefore, for this method, we compare the results reported in the original publication.

From Table 6.3, we see that the proposed method reports the best performance among all prior work, except NP-MSSG. Table 6.3 shows that using a sense-tagged corpus helps in learning better word representations by significantly outperforming (Fisher transformation at $p < 0.05$) the sense-insensitive corpus-based and joint word representations methods. However, from Table 6.3, we can see that NP-MSSG reports the best performance among other methods, which shows an advantage of cluster-based models of capturing the senses in this task.

6.4.4 Sentiment Analysis

The sentiment analysis task is one of the extrinsic evaluation tasks widely used to evaluate the word representation learning methods [9, 51, 129, 135]. In principle, extrinsic evaluations measure how well a word embedding model performs in a specific task. As such, in this section, we used the sentiment analysis task to further evaluate the proposed SAWR method and prior work. Detailed description of the task is provided in Section 6.4.4.1. Section 6.4.4.2 then goes on to highlight the benchmark datasets used in this task. The section is then concluded with the evaluation results (Section 6.4.4.3).

6.4.4.1 Task Description

In this task, the sentiment analysis is modelled as a binary sentiment classification task, where a classifier is trained using short-texts (e.g. customer reviews) labelled as *positive* or *negative* to indicate sentiment. To evaluate the word representations in this task, each short-text is represented by the centroid of the representations (learnt by a particular word representations method) of its words. In our experiment, we train a binary logistic regression classifier with the training data portion of each dataset described below and measure the classification accuracy using the corresponding test data portion.

6.4.4.2 Benchmark Datasets

We used the following widely used four binary sentiment analysis datasets to solve the sentiment classification task: Customer Reviews dataset (CR) [70] (1494 instances, 925 positive and 569 negative), Stanford Sentiment Treebank (TR) [133] (1806 test instances, 903 positive and 903 negative), Movie Reviews dataset (MR) [117] (10662 instances, 5331 positive and 5331 negative), and the Subjectivity dataset (SUBJ) [116] (10000 instances, 5000 positive and 5000 negative).

Method	CR	TR	MR	SUBJ
CBOW	72.81	73.31	67.40	82.35
Skipgram	76.07	72.87	69.41	83.55
GloVe	76.17	73.25	**70.40**	85.10
JointReps	75.83	73.62	70.49	84.70
Retro(Skipgram)	**78.29**	73.70	68.9	84.15
MSSG	75.53	72.03	69.13	84.98
NP-MSSG	73.82	70.34	68.52	84.05
SAWR	76.33	**73.75**	69.26	**85.15**

Table 6.4: Accuracy performance of the proposed SAWR in comparison with prior work evaluated on short-text sentiment classification datasets.

6.4.4.3 Results

Table 6.4 shows the result of the proposed SAWR against other methods on the sentiment analysis task. Overall, from Table 6.4, we can see that the proposed method reports the best performance results for two of the four benchmark datasets. In particular, the proposed method obtains the best results on TR and SUBJ, whereas Retro(Skipgram) and GloVe obtains the best performance on CR and MR respectively. Table 6.4 supports the conclusion drawn from Table 6.3 that it is beneficial to consider a sense-labelled corpus to obtain better word representations. For example, in SUBJ, which is among the largest short-text sentiment classification datasets, the proposed method reports the highest performance. Moreover, the proposed method significantly improves over the sense-sensitive MSSG and NP-MSSG on both TR and CR.

6.5 Summary

In this chapter, we addressed the polysemy problem and proposed the Sense-Aware Word Representations (SAWR) a method for jointly learning the word and sense representations using both an unlabelled corpus and a sense-tagged corpus. The purposed SAWR method learns the representations of words and their senses in the same lower-dimensional vector space. We conduct several sets of experiments to evaluate the word and sense representations learnt by SAWR. We initiate the evaluation tasks by creating a *pseudo sense-labelled* corpus by replacing either two or four words by a unique identifier to generate arbitrarily large sense-labelled corpus which enables us to verify the ability of SAWR to learn the sense representations. Our experimental results on this dataset show that SAWR can indeed learn word representations that are sensitive to the different senses appearing in the dataset. We then use the learnt word representations to compute the semantic similarity between two words for word-pairs that have been rated by humans, which show us that the proposed method learns accurate word representations by modelling senses. Finally, we use the word representations learnt by the proposed method to represent textual reviews on several benchmark datasets to solve the sentiment analysis task. Those experiments reveal that by incorporating unlabelled data, we can indeed learn better word representations that are sensitive to the word senses compared to what we would get if we had used only labelled data, which is encouraging given the abundance of unlabelled text corpora.

In the next chapter, the thesis is concluded with a summary of the thesis, an overview of the main findings and contributions, and some potential future work directions.

Chapter 7

Conclusion

7.1 Introduction

This chapter presents a conclusion to the work described in this thesis. The chapter begins with Section 7.2, in which an overall summary of the thesis is presented. Next, an overview of the main findings and contributions of the thesis with respect to the main research question and the subsidiary questions is reported in Section 7.3. Finally, in Section 7.4, some suggested and potential future directions based on the work described in the thesis are presented.

Before going on with the chapter sections, it is worth briefly summarising the main contributions of the thesis:

1. A joint word representations learning for additional evidence (**JointReps**) approach. The KB's knowledge in JointReps is combined with the corpus as relational constraints that must be satisfied by the learnt word representations.

2. Two KB expanding approaches, Nearest Neighbour Expansion (**NNE**) and Hedged Nearest Neighbour Expansion (**HNE**), that expand the KB from the corpus co-occurrence statistics to enhance the learning process of JointReps.

3. A fine-tuning joint word representations learning approach for hierarchical information (**HWR**). The proposed **HWR** used the hypernym relations that exist between words in a KB and the contextual information in a corpus to encode the hierarchical structure between words.

4. An evaluation task, namely **hierarchical path prediction**, with a benchmark dataset to proposed to evaluate any fine-tuned word representations for hierarchical information.
5. An evaluation task, referred to as **word decomposition**, introduced to understand the compositional structure between words.
6. A joint sense-aware word representations learning (**SAWR**) approach that utilised unlabelled and sense-labelled corpora.

7.2 Summary of Thesis

The thesis commenced with an introductory chapter, Chapter 1, in which the overview, motivations, research questions and the main contributions of the thesis are presented. The overall theme of the thesis is the investigation and exploration of joint approaches that combine text corpora and KBs to improve the overall process of learning word representations.

Recent years have witnessed a substantial abundance of textual data. As such, the primary motivation for learning word representations is to ease the understanding of natural languages for NLP systems from textual data. This, in turn, has led to the increased necessity for developing structures capable of representing textual data in a machine-understandable way. As a result, the representation of word meanings through linear algebraic structures (e.g. vectors) has arisen as an essential task in NLP.

In the literature, text corpora and KBs are two types of resources that have been widely used for learning word representations. The distributional hypothesis, which suggests that the meaning of words can often be guessed from the co-occurring nature of words in a corpus, has been used successfully for learning distributional word representations. This means that the semantic representation of a word can be represented by a vector whose dimensions correspond to its co-occurring context words. Therefore, given an adequately large text corpus, we can construct a semantic representations for words. On the other side, hand-coded KBs offer a different way to represent the meanings of words. With KBs, the linguistic properties and the semantic relations of words are exploited to learn the word representations.

Using either a text corpus or a KB, as the sole resource for learning the word representations, is usually associated with several shortcomings. For example, the word representations learnt by corpus-based approaches do not ensure that the semantic relations existing between words will be captured. In fact, it has already been shown in the literature, as discussed in Chapter 1, that corpus-based approaches suffer in different situations. For example, in estimating the strength of the semantic relationships between words accurately or in learning relation-specific representations. Similarly, an expensive manual effort will always be needed to produce KBs and then to exploit them to uncover representative features. However, combining the two types of resources provides complementary strengths when learning word representations. Thus, the aim of this thesis was to explore joint approaches that combine the text corpora and KBs to improve the learnt word representations.

The complementary strengths of combining text corpora and KBs for learning word representations are prominent on several occasions. For example, for distributional techniques to accurately represent the meaning of words from co-occurrence statistics in a corpus, plenty of occurrences for each word is required. This can be problematic with rare occurrence words. Accordingly, there is a need for additional evidence to support the rare co-occurrences. To illustrate the necessity for such additional evidence, let us assume that the rare occurring words are two synonyms. Because they rarely occur in the corpus, it will be difficult to accurately estimate the strength of their relationship. However, such synonyms would be explicitly defined in a KB. As such, the KB could be used to strengthen the similarity between those rare words. Another occasion in which the complementary strengths of a corpus and a KB is also notable is when we want to fine-tune the word representations for a particular semantic relation. For instance, the hierarchical information existing among words has been found, to some extent, encoded in word representations learnt from a corpus. KBs can further assist in encoding the hierarchical information between words by using their explicit defined hierarchical relations. Moreover, a commonly associated limitation of corpus-based word representations is their inability to handle the polysemy problem. In KBs, word senses are explicitly defined, thus can be used to disambiguate the corpus, and learn sense-aware word representations.

Consequently, this thesis sought to leverage the complementary strengths of both resources, text corpora and KBs, and proposed joint approaches for learning word representations. More specifically, three approaches were proposed:

1. The joint representation learning for additional evidence (JointReps) approach, which combined a KB into the learning process with a corpus to provide additional evidence.

2. The joint hierarchical word representation (HWR) approach that fine-tuned the word representations to encode the hierarchical structure between words.

3. The joint sense aware word representations (SAWR) learning approach that utilised unlabelled and sense-labelled (KB's-linked senses) corpora.

In Chapter 2, general background for the concepts of representation learning and word representation along with an overview of the work related to this thesis were presented. The chapter commenced with a discussion of the fundamentals of representation learning and the traditional method of data representation. This was followed by an introduction to the word representation learning and the way in which the distributional hypothesis and the vector space models (VSMs) form its backbone. The ways KBs can be leveraged for learning word representations were then discussed. The chapter next reviewed related work concerning word representations learning. The related work were categorised as being: (i) corpus-based, (ii) KB-based or (iii) joint approaches. The main focus was on the third category, which is the work that most closely aligns with this thesis. More specifically, joint approaches for (i) additional evidence, (ii) fine-tuning and (iii) word disambiguation, were thoroughly reviewed.

Chapter 3 then introduced the first joint approach proposed in the thesis. In particular, the *Joint Representation Learning for Additional Evidence* (JointReps) was introduced. JointReps utilised KB's semantic relations as *additional evidence* alongside the co-occurrence distribution in a corpus to enhance the learnt word representations. To this end, JointReps used the corpus to define a learning count- and prediction-based objective subject to the relational constraints derived from the KB. The proposed approach was evaluated in two main tasks, word similarity measurement and word analogy prediction using a wide range of benchmark datasets. The experimental results demonstrated that JointReps improved the accuracy of the word representations learnt. It outperformed a corpus-only baseline and reported an improvement over several previously proposed methods that incorporated corpora and KBs for additional evidence.

In Chapter 4, an enhancement of the JointReps approach was proposed. Specifically, the incorporation process of the KB relations into the joint learning was enhanced. The chapter addressed the lack of constraints that can be derived from the KB for each word-pair in the corpus. Towards that end, *Nearest Neighbour Expansion* (NNE) and *Hedged Nearest Neighbour Expansion* (HNE) methods were proposed for expanding the KB from the corpus co-occurrence information. The reported evaluation indicated that JointReps with NNE and HNE showed an improvement over the *Static Knowledge Base* (SKB) baseline. The empirical experiments also showed that JointReps with NNE and HNE demonstrated consistent improvements with a variety of resource sizes.

Chapter 5 presented the *joint Hierarchical Word Representation* (HWR) approach, the second joint approach proposed in this thesis. The HWR approach fine-tuned the word representations to encode the hierarchical structure between words. Previous work on learning hierarchical word representations concentrated on word-pair hypernym relations. The HWR instead exploited the full hierarchical paths in the KB. The HWR also used a hybrid of count- and prediction-based objective for incorporating the corpus into the joint learning. The reported evaluation on a range of standard tasks (e.g. supervised hypernym detection and graded lexical entailment measurement) demonstrated that the word representations learnt by the HWR approach were successfully able to encode the hierarchy. The chapter also presented two novel evaluation tasks, hierarchical path prediction and word decomposition. These were proposed in order to further evaluate the hierarchical word representations. In both tasks, the HWR approach also reported promising results to reflect the hierarchical structure in the learnt representations.

Chapter 6 then went to address the polysemy problem and introduced the third joint approach of this thesis. In particular, the chapter presented the *Sense Aware Word Representations* (SAWR) approach. The SAWR approach jointly learned the word and sense representations using unlabelled and sense-labelled corpora. It embedded the words and their senses in the same lower-dimensional embedding space. The SAWR attempted to see whether an unlabelled corpus could help the process of learning sense representations when it combined with a sense-labelled corpus. The experimental results revealed that SAWR could indeed learn word representations that were aware of the different senses of a word. The chapter also showed that more semantically accurate word representations could be learnt by modelling the senses.

7.3 Main Findings and Contributions

This section revisits the main research question and the associated subsidiary research questions presented in Chapter 1 (Section 1.3). The section addresses these questions with respect to the “main findings” of the work presented in this thesis. Thus, the section is organised in light of each subsidiary research question before returning to the main research question.

1. **Additional Evidence**: “*What is the most appropriate mechanism to incorporate a KB with a corpus to provide additional evidence to the corpus distributional information?*”. Initially, for the corpus distributional information, instead of relying on either count-based or prediction-based objectives, a hybrid objective which combined the advantages of the two was leveraged. The question of which mechanism could best be used to incorporate the KB’s knowledge as additional evidence was then addressed by introducing such knowledge as relational constraints. Such relation constraints must be satisfied by the learnt word representation. For that purpose, an objective was defined to consider a three-way co-occurrence rather than only a two-way co-occurrence. That is, the co-occurrence between the words in the corpus and the semantic relations existing between them in the KB. The two objectives were then formulated as a joint objective modelling the proposed JointReps approach.

   According to the conducted evaluation, presented in Chapter 3, the mechanism of jointly learning the word representations was found to be significantly effective for providing additional evidence from the KB. It helped to learn semantically and syntactically better word representations as compared to using only the corpus. In addition, the joint mechanism of combining the two types of resources was also found to be more effective than the retrofitting mechanism. That is because the rich information in the

KBs was being utilised during the learning process rather than during post-processing. Moreover, the experimental results established that is wan not only the synonym relations that were helpful as additional evidence. In fact, other semantic relations such as hypernyms, hyponyms, meronyms and holonyms were also useful for learning better word representations.

2. **Dynamic KB Expansion**: "*Can we utilise the corpus co-occurrence statistics to compensate for the limited number of entries for words in a KB? If so, what is the appropriate mechanism to use the corpus co-occurrence information to expand a KB?*". The JointReps considered each word-pair co-occurring in a corpus and sought the semantic relations between those words in a KB to enhance the learnt word representations. However, it was found that only around one-eighth of the words in the corpus existed in the KB. The proposed solution was then to introduce two expansion methods, NNE and HNE.

   The NNE and HNE dynamically expanded the KB using the information extracted from the corpus. The NNE expanded the KB based on the co-occurrence counts between words in the corpus. The HNE operated similarly to NNE but filtered the co-occurrence noise in the corpus before dynamically updating the KB. More specifically, NNE worked based on the assumption that if two words frequently co-occur in the corpus, then there is a strong possibility that those two words are semantically related. However, it was observed that the NNE assumption yielded some noises as expansion candidates (e.g. hub words). HNE addressed this drawback by requiring the expansion candidates to satisfy some conditions prior to the expansion process. For example, it required the candidate words to be among the nearest neighbours of two words and for the two words to be recorded with a semantic relation in the KB.

   The conducted experimental analysis established that the expansion by NNE and HNE showed promise with respect to learning more accurate word representations. Besides, the qualitative analysis revealed that HNE was able to successfully eliminate some potential noisy expansion words. Moreover, repeatedly expanding the KB using NNE and HNE was observed to further improve the learnt word representations. Furthermore, the quality of the representations was noted to increase with the amount of corpus and KB data.

3. **Fine-Tuned Representations**: "*How can a KB and a corpus be best combined to fine-tune word representations to a target task or to represent a particular semantic relation? What is the best mechanism to extract KB and corpus data for that purpose?*". The challenge of finding the appropriate technique to fine-tune the word representations was addressed by proposing an approach that fine-tuned the representations for hierarchical information. In contrast to previous works which mainly focused on pairwise hypernym relations, the proposed approach addressed this by utilising the full hierarchical path of words from the KB.

   The proposed method was built around the assumption that the words in the KB are arranged in a hierarchical order. Thus, some of the information contained in a leaf node (hyponym word) could be inferred from its parent nodes (hypernym words) that fall along the paths. More specifically, an objective was defined to learn the representation of a leaf node as the sum of its parents' representations. To incorporate the corpus, another hybrid corpus- and prediction-based objective was defined that examined the co-occurrences in the corpus between the hyponym word and each of its hypernyms that appeared in the path. The two objectives were then combined to formalise the final objective of the joint HWR approach. The reported evaluation results showed that the proposal of using the full hierarchical path of hypernyms is indeed beneficial. It not only gave a better understanding of the hierarchy but also was useful for a pairwise hypernymy identification.

4. **Evaluation of HWR**: "*How can we properly evaluate the enhancement brought by the joint approach on the fine-tuned word representations?*". The experimental results in Chapter 5 revealed that the proposed HWR approach achieved an improvement in reflecting the hierarchy. In particular, it reported an improvement over corpus-based, non-fine-tuning, and fine-tuning approaches on several standard tasks. However, those tasks (e.g. supervised/unsupervised hypernymy detection and graded lexical entailment) provided only a partial evaluation concerning hierarchy. Because the benchmark

datasets in those tasks were mainly focused on the hypernymy between two words, disregarding the full hierarchical path.

The suggested solution was to evaluate the word representations for their ability to capture the full hierarchical information available in a KB. To that end, a novel evaluation task, hierarchical path prediction, along with a benchmark dataset, were proposed. The task was to predict a hyponym word that fits best to complete a word-missing hierarchical path. Moreover, to further go beyond evaluating the HWRs on pairwise-focused tasks, another task was proposed. The task, word decomposition, was designed to examine how a word's meaning can be related to the meanings of its parent concepts. The proposed task was to express the HWRs of a word as the linearly-weighted combination of a set of given words. Both tasks showed that the proposed method HWR had the capability to encode hierarchy into the learnt word representations. Furthermore, a significant drop in the performance of previously proposed approaches for fine-tuning the representations was observed in the path prediction task as compared to the performance in the pairwise-focused tasks. As such, a concern about whether such standard pairwise tasks are the most appropriate tasks to evaluate the HWRs was raised.

5. **Sense Representations**: "*How can sense-aware word representations best be learnt from sense related information available in a KB and contextual clues in the corpus?*". Two main lines could be identified in the literature that have addressed the polysemy problem when learning word representations. Either using a corpus with a KB specifying the word senses, or using a word-sense disambiguation tool that generates an automatic sense-labelled corpus. No prior work on learning sense-aware word representations has attempted to address the issue of polysemy by combining unlabelled and sense-labelled corpora. As such, the proposed solution was to consider the two.

   In particular, the proposed method used a large collection of unlabelled texts and a comparatively smaller collection of sense-labelled sentences to learn both word and sense representations simultaneously. To this end, several objectives were defined and later combined into a single linearly-weighted objective that modelled the joint SAWR approach. It required the word representations to be able to predict the co-occurrences of both words and senses associated with the words in a context. Evaluation (qualitative and quantitative) results indicated that it was indeed helpful to incorporate the unlabelled data. It helped to learn word representations that are sensitive to the word senses. Those results were encouraging because such unlabelled data is abundant.

Returning to the main research question:

"*Is it possible to enhance the word representations by jointly incorporating text corpora and KBs into the word representations learning process? If so, what are the aspects of word meaning that can be enhanced by combining those two types of resources?*"

From the preceding, several joint approaches were proposed. In particular, (i) joint word representations learning for additional evidence (JointReps), (ii) joint hierarchical word representations (HWR) and (iii) joint sense-aware word representations (SAWR). The experiments and the analyses conducted on those approaches showed that it is indeed possible to enhance the word representations learning by incorporating text corpora and KBs. More specifically, each of the joint approaches proposed in the thesis provided an enhancement to the learnt word meaning representations from different aspects. The JoinReps approach improved the overall semantic representation of words by injecting additional evidence that further pulled semantically similar words closer together. The HWR method fine-tuned the representations to enhance and reflect the hierarchical meaning existing between words. The SAWR approach enhanced the sense part of the word meaning representations by considering not only the single meaning of words but also their different senses.

7.4 Future Work

The work presented in this thesis has proposed several joint approaches for learning word representations by combining text corpora and KBs. It has demonstrated that the joint mechanism can effectively achieve a better semantic representation of words. Despite the results achieved, some enhancements and improvements can be adopted. This final section suggests a number of potential directions for future work as detailed below:

1. **Integrating Antonyms into JointReps**: As discussed in detail in Chapter 1 (Section 1.2), one of the strengths of combining a corpus and a KB for learning word representations was evident as a potential resolution to the issue of antonyms in the corpus. The proposed JointReps has implicitly addressed this issue. In particular, JointReps incorporated the synonym relations, and hence further pulled the synonymy closer together in the embeddings space. However, the JointReps objective was not explicitly defined to consider the antonym relations. In other words, JointReps has not enforced the antonyms to be far from each other in the embedding space. Thus, investigation on integrating an objective to explicitly model antonyms into JointReps is deemed to be a fruitful avenue for future work.

2. **Evaluating in Downstream NLP Applications**: As noted in the foregoing chapters, the quality of the learnt word representations could be evaluated either: (i) extrinsically, in which the representations are used as input features in a downstream NLP task (e.g. short-text classification, unsupervised hypernym detection, etc.) or (ii) intrinsically as independent of any specific downstream NLP task (e.g. word similarity measurement, word analogy prediction, etc.). Although all the proposed joint approaches showed promising results in both types of evaluation, the number of extrinsic evaluation tasks was limited. Some recent studies have suggested that evaluating the representations in downstream NLP tasks has certain advantages over intrinsic evaluation [9, 53]. Therefore, a broader evaluation that applies the learnt word representations in more downstream NLP applications (e.g. sentiment analysis, NER, metaphor detection and question answering) appears to be a fruitful direction for future research.

3. **General Framework**: Despite the success of the proposed joint approaches in enhancing different aspects of the semantic representations of words, these approaches operated independently. For example, the JointReps approach aimed at enhancing the overall semantic representations using additional evidence from the KB. However, JointReps did not consider the different senses

of words or the hierarchy. Similarly, the HWR approach attempted to reflect the hierarchical structure existing among words in the learnt representations. Nevertheless, the HWR did not explicitly look into the different senses of words. Recent work has shown that it is useful to consider all those aspects simultaneously while learning word representations [140]. Consequently, a synthesis of the three proposed approaches under a general framework seems to be a promising direction for future work.

4. **Contextualised Word Representations**: Most recently, the literature has witnessed a new line of work for learning word representations that has received a great deal of attention. Namely, deep neural language models such as Embeddings from Language Models (ELMo) [120] and Bidirectional Encoder Representations from Transformers (BERT) [41] approaches that learn contextualised word representations. Such methods learn word vectors that are sensitive to the context in which the words appear in. The aim here is to capture word semantics in different contexts to address the polysemy problem and the context-dependent nature of words. The contextualised representations, when integrated with task-specific architectures, have achieved state-of-the-art results in a wide range of NLP tasks, ranging from question answering to NER. The contextualised representations are unsupervisedly learned purely from large text corpora. As such, learning contextualised word representations jointly from a corpus and a KB appear to be another possible direction for future research.

5. **Utilising Further Corpora and KBs**: The joint approaches that have been proposed in this thesis used ukWaC as a corpus and WordNet as a KB. However, as noted throughout the thesis, all these joint approaches are not limited to any particular corpus, KB, domain or language. For example, JointReps and HWR can be applied to any corpus, irrespective of the domain or the language, provided that the co-occurrence statistics between words are attainable. Similarly, any KB that provides pairwise relationships or hierarchical paths between words can be used as a KB with JointReps or HWR respectively. As such, in future, we plan to apply the proposed joint approaches using corpora and KBs from different domains and languages. One promising direction is to use JointReps and HWR with a biomedical cor-

pus such as Medline[1] and a biomedical KB such as Snomed-CT[2]. Moreover, applying the proposed joint approaches on different languages is another promising direction for future work. For example, there are several available Arabic corpora such as KSUCCA [2] and ArTenTen [8] which can be combined with the Arabic WordNet [47] using JointReps or HWR for jointly learning Arabic word embeddings.

[1] https://www.ncbi.nlm.nih.gov/pubmed

[2] http://www.snomed.org/snomed-ct/

Bibliography

[1] Felipe Almeida and Geraldo Xexéo. Word embeddings: A survey. *arXiv preprint arXiv:1901.09069*, 2019.

[2] Maha Alrabiah, A Al-Salman, and ES Atwell. The design and construction of the 50 million words ksucca. In *Proceedings of WACL'2 Second Workshop on Arabic Corpus Linguistics*, pages 5–8, 2013.

[3] Mohammed Alsuhaibani and Danushka Bollegala. Joint learning of sense and word embeddings. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC)*, 2018.

[4] Mohammed Alsuhaibani, Danushka Bollegala, Takanori Maehara, and Ken-ichi Kawarabayashi. Jointly learning word embeddings using a corpus and a knowledge base. *PLOS ONE*, 13(3):1–26, 03 2018.

[5] Mohammed Alsuhaibani, Takanori Maehara, and Danushka Bollegala. Joint learning of hierarchical word embeddings from a corpus and a taxonomy. *Automated Knowledge Base Construction (AKBC)*, 2019.

[6] Robert A. Amsler. Lexical knowledge bases. In *Proceeding of the 10th International Conference on Computational Linguistics (COLING) and 22nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 458–459, 1984.

[7] Tuan Luu Anh, Yi Tay, Siu Cheung Hui, and See Kiong Ng. Learning term embeddings for taxonomic relation identification using dynamic weighting neural network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 403–413, 2016.

[8] Tressy Arts, Yonatan Belinkov, Nizar Habash, Adam Kilgarriff, and Vit Suchomel. artenten: Arabic corpus and word sketches. *Journal of King Saud University-Computer and Information Sciences*, 26(4):357–371, 2014.

[9] Amir Bakarov. A survey of word embeddings evaluation methods. *arXiv preprint arXiv:1801.09536*, 2018.

[10] Collin F Baker, Charles J Fillmore, and John B Lowe. The berkeley framenet project. In *Proceedings of the 17th International Conference on Computational Linguistics (COLING)*, pages 86–90, 1998.

[11] Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 23–32, 2012.

[12] Marco Baroni, Raffaella Bernardi, and Roberto Zamparelli. Frege in space: A program of compositional distributional semantics. *Linguistic Issues in Language Technology*, 9:5–110, 2014.

[13] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 238–247, 2014.

[14] Marco Baroni and Alessandro Lenci. How we blessed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10. Association for Computational Linguistics, 2011.

[15] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of International Conference on Machine Learning (ICML) workshop on unsupervised and transfer learning*, pages 17–36, 2012.

[16] Yoshua Bengio, Aaron Courville, and Pierre Vincent. Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1798–1828, 2013.

[17] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research (JMLR)*, 3(Feb):1137–1155, 2003.

[18] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 153–160, 2007.

[19] Or Biran and Kathleen McKeown. Classifying taxonomic relations between pairs of wikipedia articles. In *Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 788–794, 2013.

[20] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics (TACL)*, 5:135–146, 2017.

[21] Danushka Bollegala, Mohammed Alsuhaibani, Takanori Maehara, and Ken-ichi Kawarabayashi. Joint word representation learning using a corpus and a semantic lexicon. In *the 30th AAAI Conference on Artificial Intelligence*, pages 2690–2696, 2016.

[22] Danushka Bollegala, Kohei Hayashi, and Ken-ichi Kawarabayashi. Learning linear transformations between counting-based and prediction-based word embeddings. *PLOS ONE*, 12(9):e0184544, 2017.

[23] Danushka Bollegala, Takanori Maehara, and Ken-ichi Kawarabayashi. Unsupervised cross-domain word representation learning. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL) and the 7th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 730–740, 2015.

[24] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. Joint learning of words and meaning representations for open-text semantic parsing. In *Artificial Intelligence and Statistics (AISTATS)*, pages 127–135, 2012.

[25] Antoine Bordes, Jason Weston, and Nicolas Usunier. Open question answering with weakly supervised embedding models. In *Joint European Confer-*

ence on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD), pages 165–180. Springer, 2014.

[26] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. *arXiv preprint arXiv:1206.6392*, 2012.

[27] Elia Bruni, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran. Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 136–145, 2012.

[28] José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. Nasari: a novel approach to a semantically-aware representation of items. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 567–577, 2015.

[29] Haw-Shiuan Chang, Ziyun Wang, Luke Vilnis, and Andrew McCallum. Distributional inclusion vector embedding for unsupervised hypernymy detection. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 485–495, 2018.

[30] Zhigang Chen, Wei Lin, Qian Chen, Xiaoping Chen, Si Wei, Hui Jiang, and Xiaodan Zhu. Revisiting word embedding for contrasting meaning. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL) and the 7th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 106–115, 2015.

[31] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29, 1990.

[32] Dan Cireşan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. *arXiv preprint arXiv:1202.2745*, 2012.

[33] Charles J Clopper and Egon S Pearson. The use of confidence or fiducial

limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

[34] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research (JMLR)*, 12:2493–2537, 2011.

[35] Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. Recognizing textual entailment: Models and applications. *Synthesis Lectures on Human Language Technologies*, 6(4):1–220, 2013.

[36] George Dahl, Marc'Aurelio Ranzato, Abdel-rahman Mohamed, and Geoffrey E Hinton. Phone recognition with the mean-covariance restricted boltzmann machine. In *Advances in Neural Information Processing Systems (NIPS)*, pages 469–477, 2010.

[37] George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pretrained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 20(1):30–42, 2011.

[38] Mark Davies. The 385+ million word corpus of contemporary american english (1990–2008+): Design, architecture, and linguistic insights. *International journal of corpus linguistics*, 14(2):159–190, 2009.

[39] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[40] Li Deng, Michael L Seltzer, Dong Yu, Alex Acero, Abdel-rahman Mohamed, and Geoff Hinton. Binary coding of speech spectrograms using a deep autoencoder. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2010.

[41] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), pages 4171–4186, 2019.

[42] Paramveer Dhillon, Dean P Foster, and Lyle H Ungar. Multi-view learning of word embeddings via cca. In *Advances in Neural Information Processing Systems (NIPS)*, pages 199–207, 2011.

[43] Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. Improving zero-shot learning by mitigating the hubness problem. *arXiv preprint arXiv:1412.6568*, 2014.

[44] L Dodds and T Scott. Bbc ontologies-the wildlife ontology. 2010.

[45] Kevin Donnelly. Snomed-ct: The advanced terminology and coding system for ehealth. *Studies in health technology and informatics*, 121:279, 2006.

[46] Chris Dyer. Notes on noise contrastive estimation and negative sampling. *arXiv preprint arXiv:1410.8251*, 2014.

[47] Sabri Elkateb, William Black, Horacio Rodríguez, Musa Alkhalifa, Piek Vossen, Adam Pease, and Christiane Fellbaum. Building a WordNet for Arabic. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, 2006.

[48] Katrin Erk. Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, 6(10):635–653, 2012.

[49] Allyson Ettinger, Philip Resnik, and Marine Carpuat. Retrofitting sense-specific word vectors using parallel text. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1378–1383, 2016.

[50] Manaal Faruqui. *Diverse Context for Learning Word Representations.* PhD thesis, Carnegie Mellon University, 2016.

[51] Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. Retrofitting word vectors to semantic lexicons.

In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1606–1615, 2015.

[52] Manaal Faruqui and Chris Dyer. Non-distributional word vector representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL) and the 7th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 464–469, 2015.

[53] Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. Problems with evaluation of word embeddings using word similarity tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 30–35, 2016.

[54] Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. Introducing and evaluating ukwac, a very large web-derived corpus of english. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google*, pages 47–54, 2008.

[55] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, z. Solan, G. Wolfman, and E. Ruppin. Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, 20:116–131, 2002.

[56] JR Firth. A synopsis of linguistic theory 1930-1955 in studies in linguistic analysis. philological society, 1957.

[57] Samah Fodeh, Bill Punch, and Pang-Ning Tan. On ontology-driven document clustering using core semantic features. *Knowledge and Information Systems (KAIS)*, 28(2):395–421, 2011.

[58] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. Ppdb: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 758–764, 2013.

[59] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings*

of the 28th Iinternational Conference on Machine Learning (ICML), pages 513–520, 2011.

[60] Josu Goikoetxea, Aitor Soroa, and Eneko Agirre. Random walks and neural network language models on knowledge bases. In *Proceedings of the 2015 conference of the North American Chapter of the Association for Computational Linguistics: Human language technologies (NAACL-HLT)*, pages 1434–1439, 2015.

[61] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[62] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 297–304, 2010.

[63] Philippe Hamel and Douglas Eck. Learning features from music audio with deep belief networks. In *The Transactions of the International Society for Music Information Retrieval (ISMIR)*, volume 10, pages 339–344, 2010.

[64] Philippe Hamel, Simon Lemieux, Yoshua Bengio, and Douglas Eck. Temporal pooling and multiscale learning for automatic annotation and ranking of music audio. In *The Transactions of the International Society for Music Information Retrieval (ISMIR)*, pages 729–734, 2011.

[65] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[66] Felix Hill, Roi Reichart, and Anna Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 2016.

[67] Geoffrey E Hinton et al. Learning distributed representations of concepts. In *Proceedings of the 8th Annual Conference of the Cognitive Science Society*, volume 1, page 12. Amherst, MA, 1986.

[68] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[69] Harold Hotelling. Relations between two sets of variates. In *Breakthroughs in statistics*, pages 162–190. Springer, 1992.

[70] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the 10th International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 168–177, 2004.

[71] Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 873–882, 2012.

[72] Zhiheng Huang, Marcus Thint, and Zengchang Qin. Question classification using head words and their hypernyms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 927–936, 2008.

[73] Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. Sensembed: Learning sense embeddings for word and relational similarity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL) and the 7th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 95–105, 2015.

[74] Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 897–907, 2016.

[75] Nancy Ide and Keith Suderman. The American national corpus first release. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)*, 2004.

[76] Sujay Kumar Jauhar, Chris Dyer, and Eduard Hovy. Ontologically grounded multi-sense representation learning for semantic vector space models. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 683–693, 2015.

[77] Richard Johansson and Luis Nieto Piña. Embedding a semantic network in a word space. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1428–1433, 2015.

[78] Tom Kenter, Alexey Borisov, and Maarten de Rijke. Siamese CBOW: Optimizing word embeddings for sentence representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 941–951, 2016.

[79] Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16(4):359–389, 2010.

[80] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.

[81] Henry Kučera and Winthrop Nelson Francis. *Computational analysis of present-day American English.* Dartmouth Publishing Group, 1967.

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

[83] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 260–270, 2016.

[84] Thomas K Landauer and Susan T Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211, 1997.

[85] Jey Han Lau, Paul Cook, Diana McCarthy, David Newman, and Timothy Baldwin. Word sense induction for novel sense detection. In *Proceedings of*

the 13th Conference of the European Chapter of the Association for Computational Linguistics (ACL), pages 591–601. Association for Computational Linguistics, 2012.

[86] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning (ICML)*, pages 1188–1196, 2014.

[87] Rémi Lebret and Ronan Collobert. Word embeddings through hellinger PCA. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 482–490, 2014.

[88] Omer Levy, Ido Dagan, and Jacob Goldberger. Focused entailment graphs for open ie propositions. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning (CoNLL)*, pages 87–97, 2014.

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

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

[91] Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 970–976, 2015.

[92] Quan Liu, Hui Jiang, Si Wei, Zhen-Hua Ling, and Yu Hu. Learning semantic word embeddings based on ordinal knowledge constraints. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL) and the 7th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 1501–1511, 2015.

[93] Kevin Lund and Curt Burgess. Producing high-dimensional semantic spaces

from lexical co-occurrence. *Behavior research methods, instruments, & computers*, 28(2):203–208, 1996.

[94] Thang Luong, Richard Socher, and Christopher Manning. Better word representations with recursive neural networks for morphology. In *Proceedings of the 17th Conference on Computational Natural Language Learning (CoNLL)*, pages 104–113, 2013.

[95] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research (JMLR)*, 9(Nov):2579–2605, 2008.

[96] Ian RA MacKay. David crystal. a dictionary of linguistics and phonetics. london: Blackwell. 1985. *Canadian Journal of Linguistics/Revue canadienne de linguistique*, 32(2):220–223, 1987.

[97] James H Martin and Daniel Jurafsky. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Pearson/Prentice Hall Upper Saddle River, 2009.

[98] Oren Melamud, Omer Levy, and Ido Dagan. A simple word embedding model for lexical substitution. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 1–7, 2015.

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

[100] Tomáš Mikolov, Anoop Deoras, Stefan Kombrink, Lukáš Burget, and Jan Černockỳ. Empirical evaluation and combination of advanced language modeling techniques. In *12th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2011.

[101] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3111–3119, 2013.

[102] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 746–751, 2013.

[103] G. Miller and W. Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28, 1998.

[104] George A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39 – 41, 1995.

[105] George A Miller, Claudia Leacock, Randee Tengi, and Ross T Bunker. A semantic concordance. In *Proceedings of the workshop on Human Language Technology (HLT)*, pages 303–308, 1993.

[106] Andriy Mnih and Koray Kavukcuoglu. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2265–2273, 2013.

[107] Abdel-rahman Mohamed, George E Dahl, and Geoffrey Hinton. Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech and Language Processing*, 20(1):14–22, 2011.

[108] Nikola Mrkšić, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young. Semantic specialization of distributional word vector spaces using monolingual and cross-lingual constraints. *Transactions of the Association for Computational Linguistics (TACL)*, 5:309–324, 2017.

[109] Roberto Navigli, Paola Velardi, and Stefano Faralli. A graph-based algorithm for inducing lexical taxonomies from scratch. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, volume 11, pages 1872–1877, 2011.

[110] Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. Efficient non-parametric estimation of multiple embeddings per

word in vector space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1059–1069, 2014.

[111] Kim Anh Nguyen, Maximilian Köper, Sabine Schulte im Walde, and Ngoc Thang Vu. Hierarchical embeddings for hypernymy detection and directionality. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 233–243, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[112] Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 454–459, 2016.

[113] Thien Huu Nguyen and Ralph Grishman. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48, 2015.

[114] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6338–6347, 2017.

[115] Yoshiki Niwa and Yoshihiko Nitta. Co-occurrence vectors from corpora vs. distance vectors from dictionaries. In *Proceedings of the 15th conference on Computational linguistics-Volume 1*, pages 304–309, 1994.

[116] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL)*, page 271, 2004.

[117] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*, pages 115–124, 2005.

[118] Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. English gigaword. *Linguistic Data Consortium*, 2011.

[119] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[120] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2227–2237, 2018.

[121] Victor Prokhorov, Mohammad Taher Pilehvar, Dimitri Kartsaklis, Pietro Lió, and Nigel Collier. Learning rare word representations using semantic bridging. *arXiv preprint arXiv:1707.07554*, 2017.

[122] M Ross Quillian. Word concepts: A theory and simulation of some basic semantic capabilities. *Behavioral science*, 12(5):410–430, 1967.

[123] Radim Rehurek and Petr Sojka. Software framework for topic modelling with large corpora. In *In Proceedings of the International Conference on Language Resources and Evaluation (LREC) 2010 Workshop on New Challenges for NLP Frameworks*, 2010.

[124] Joseph Reisinger and Raymond J Mooney. Multi-prototype vector-space models of word meaning. In *Proceedings of the 2010 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 109–117. Association for Computational Linguistics, 2010.

[125] Douglas LT Rohde, Laura M Gonnerman, and David C Plaut. An improved model of semantic similarity based on lexical co-occurrence. *Communications of the ACM*, 8(627-633):116, 2006.

[126] Sascha Rothe and Hinrich Schütze. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL) and the 7th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 1793–1803, 2015.

[127] H. Rubenstein and J.B. Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8:627–633, 1965.

[128] Ivan Sanchez and Sebastian Riedel. How well can we predict hypernyms from word embeddings? a dataset-centric analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, volume 2, pages 401–407, 2017.

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

[130] Frank Seide, Gang Li, and Dong Yu. Conversational speech transcription using context-dependent deep neural networks. In *12th Annual Conference of the International Speech Communication Association*, 2011.

[131] John Sinclair. *Corpus, concordance, collocation.* Oxford University Press, 1991.

[132] Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 151–161, 2011.

[133] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642, 2013.

[134] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL) and the 7th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 1556–1566, 2015.

[135] Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. Evaluation of word vector representations by subspace alignment. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2049–2054, 2015.

[136] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 384–394, 2010.

[137] Peter D Turney. Mining the web for synonyms: Pmi-ir versus lsa on toefl. In *European conference on machine learning*, pages 491–502. Springer, 2001.

[138] Peter D. Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of Aritificial Intelligence Research*, 37:141 – 188, 2010.

[139] Ivan Vulić, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. Hyperlex: A large-scale evaluation of graded lexical entailment. *Computational Linguistics*, 43(4):781–835, 2017.

[140] Ivan Vulić and Nikola Mrkšić. Specialising word vectors for lexical entailment. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1134–1145, 2018.

[141] Ekaterina Vylomova, Laura Rimell, Trevor Cohn, and Timothy Baldwin. Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1671–1682, 2016.

[142] Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller. Learning to distinguish hypernyms and co-hyponyms. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, pages 2249–2259, 2014.

[143] Ludwig Wittgenstein. *Philosophical investigations.* John Wiley & Sons, 2009.

[144] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[145] Chang Xu, Yalong Bai, Jiang Bian, Bin Gao, Gang Wang, Xiaoguang Liu, and Tie-Yan Liu. Rc-net: A general framework for incorporating knowledge into word representations. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM)*, pages 1219–1228. ACM, 2014.

[146] Dong Yu and Li Deng. Deep learning and its applications to signal and information processing [exploratory dsp]. *IEEE Signal Processing Magazine*, 28(1):145–154, 2010.

[147] Liang-Chih Yu, Jin Wang, K. Robert Lai, and Xuejie Zhang. Refining word embeddings for sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 534–539, 2017.

[148] Mo Yu and Mark Dredze. Improving lexical embeddings with semantic knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 545–550, 2014.

[149] Zheng Yu, Haixun Wang, Xuemin Lin, and Min Wang. Learning term embeddings for hypernymy identification. In *24th International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

[150] Will Y. Zou, Richard Socher, Daniel Cer, and Christopher D. Manning. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1393–1398, 2013.