UNIVERSITY OF LIVERPOOL

A Study on Learning Representations for Relations Between Words

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

Faculty of Science and Engineering
School of Electrical Engineering, Electronics and Computer Science
Department of Computer Science

by

Huda Hakami

May 2020

Acknowledgements

Thank God for his grace to accomplish my PhD thesis. I would like to thank everyone who stood beside me and made it easier for me to complete the PhD study.

First and foremost, my great thank and praise goes to my supervisor Prof. Danushka Bollegala who support me in various aspects and truly provided unlimited supervision for my research. I learnt from him a lot about scientific research methodologies. Honestly, I consider my self lucky for getting this chance to study under his supervision.

I also wish to thank my second supervisor Prof. Yannis Goulermas and the IPAP members, Prof. Frans Coenen and Dr. Vitaliy Kurlin, for their feedback throughout different stages of my research. I would like to thank the members in the Natural Language Processing research group at Computer Science Department (NLP@Liv) lead by my supervisor Prof. Danushka as I have learnt a lot from their input. I was also a member in the Data Mining and Machine Learning (DMML) research group lead by my supervisor, thanks to all the group members. Thanks to my friends and colleagues for their continuous support.

I cannot forget thanking my family, my parents, sisters and brothers who continuously support me to achieve my goal. Especial thanks to my loved husband, Saeed. I appreciate everything you always do to make me strong and smile. Thank you, Saeed, for being engaged, listening to me and sharing ideas. I would never become who I am today if you weren't there throughout my study journey in Liverpool motivating me with your kindness and generosity. My little angels, Mohammad and Faisal, who have always been a source of happiness for me.

This thesis would not have been possible without the funding for my study at the University of Liverpool. My thanks also go to the Saudi Arabia Ministry of Education, Taif University in particular, who gave me the scholarship to study the PhD at the University of Liverpool. I extend my thanks to the department of computer science at the University of Liverpool, which provided me with the necessary capabilities to complete the research properly. Finally, I want to say that I spent the most beautiful years of my life while studying in Liverpool.

Abstract

Reasoning about relations between words or entities plays an important role in human cognition. It is thus essential for a computational system which processes human languages to be able to understand the semantics of relations to simulate human intelligence. Automatic relation learning provides valuable information for many natural language processing tasks including ontology creation, question answering and machine translation, to name a few. This need brings us to the topic of this thesis where the main goal is to explore multiple resources and methodologies to effectively represent relations between words.

How to effectively represent semantic relations between words remains a problem that is underexplored. A line of research makes use of relational patterns, which are the linguistic contexts in which two words co-occur in a corpus to infer a relation between them (e.g., X *leads to* Y). This approach suffers from data sparseness because not every related word-pair co-occurs even in a large corpus. In contrast, prior work on learning word embeddings have found that certain relations between words could be captured by applying linear arithmetic operators on the corresponding pre-trained word embeddings. Specifically, it has been shown that the vector offset (expressed as PairDiff) from one word to the other in a pair encodes the relation that holds between them, if any. Such a compositional method addresses the data sparseness by inferring a relation from constituent words in a word-pair and obviates the need of relational patterns.

This thesis investigates the best way to compose word embeddings to represent relational instances. A systematic comparison is carried out for unsupervised operators, which in general reveals the superiority of the PairDiff operator on multiple word embedding models and benchmark datasets. Despite the empirical success, no theoretical analysis has been conducted so far explaining why and under what conditions PairDiff is optimal. To this end, a theoretical analysis is conducted for the generalised bilinear operators that can be used to measure the relational distance between two word-pairs. The main conclusion is that, under certain assumptions, the bilinear operator can be simplified to a linear form, where the widely used PairDiff operator is a special case.

Multiple recent works raised concerns about existing unsupervised operators for inferring relations from pre-trained word embeddings. Thus, the question of whether it is possible to learn better parametrised relational compositional operators is addressed in this thesis. A supervised relation representation operator is proposed using a non-linear neural network that performs relation prediction. The evaluation on two benchmark datasets reveals that the penultimate layer of the trained neural network-based relational predictor acts as a good representation for the relations between words. Because we believe that both relational patterns and word embeddings provide complementary information to learn relations, a self-supervised context-guided relation embedding method that is trained on the two sources of information has been proposed. Experimentally, incorporating relational contexts shows improvement in the performance of a compositional operator for representing unseen word-pairs.

Besides unstructured text corpora, knowledge graphs provide another source for relational facts in the form of nodes (i.e., entities) connected by edges (i.e., relations). Knowledge graphs are employed widely in natural language processing applications such as question answering and dialogue systems. Embedding entities and relations in a graph have shown impressive results for inferring previously unseen relations between entities. This thesis contributes to developing a theoretical model to infer a relationship between the connections in the graph and the embeddings of entities and relations. Learning graph embeddings that satisfy the proven theorem demonstrates efficient performance compared to existing heuristically derived graph embedding methods. As graph embedding methods generate representations for only existing relation types, a relation composition task is proposed in the thesis to tackle this limitation.

Contents

List of Figures

List of Tables

Abbreviations

The abbreviations used throughout this thesis are listed below.

BATS	Bigger Analogy Test Set
CBOW	Continuous Bag-of-Word embeddings
CGRE	Context-Guided Relation Embeddings
GloVe	Global Vector predictions
HSC	Hierarchical Sparse Coding
KG	Knowledge Graph
KGE	Knowledge Graph Embedding
KL	Kullback–Leibler divergence
LDA	Latent Dirichlet Allocation
LRA	Latent Relational Analysis
LSA	Latent Semantic Analysis
LSTM	Long Short Term Memory
MnnPL	Multi-class neural network Penultimate Layer
NLP	Natural Language Processing
NLRA	Neural Latent Relational Analysis
NMF	Nonnegative Matrix Factorisation
PMI	Point-wise Mutual Information
PPMI	Positive Point-wise Mutual Information
POS	Parts-Of-Speech
PPMI	Positive Point-wise Mutual Information
RelWalk	Relational Walk
SAT	Scholastic Aptitude Test
SG	Skip-Gram
SGD	Stochastic Gradient Descent
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TransE	Translating Embedding
VSM	Vector Space Model

Glossary

Semantic relations. Associations that exist between the meanings of words (Hypernym, Meronym, capital-of, CEO, etc.).

Syntactic relations. In the context of the study in this thesis, syntactic relations refer to associations between different morphological forms of words (e.g., between *apply* and *applied* or between *aware* and *unaware*). Syntactic relations are also called morphological relations.

Word analogies. A way of expressing a relationship between words and is formally written as $a : b :: c : d$ (read a is to b as c is to d).

Analogy questions. Test questions that require reasoning about relations. One form of analogy question is where you are given a word-pair and you are asked to choose another word-pair with the same relationship. Another form is filling a missing word given two word-pairs (i.e., $a : b :: c :?$).

Analogous word-pairs. A relation in the pair (a, b) is analogous to that in (c, d).

Non-analogous word-pairs. A relation in the pair (a, b) is different from that in (c, d).

Relational similarity. Is a similarity between two pairs of words obtained by comparing a relation that holds between the two words in a first pair with that of a second pair.

PairDiff. A method of generating a vector offset of a pair of words (a, b) by subtracting pre-trained word embedding vectors $(\boldsymbol{b} - \boldsymbol{a})$, which in turn can be used to predict relations between words.

Word embeddings (or representations). Real-valued low dimensional vectors that capture semantic and syntactic properties of the words in a vocabulary.

Knowledge graph embeddings. Representations of entities and relations in a given graph, which enable predicting missing links in the graph.

Relational patterns. Linguistic contexts that connect two related entities in a text corpus.

Pattern-based approach. A popular approach for representing relations between words relying on the patterns extracted from linguistic contexts in which the pairs of words co-occur in a text corpus, e.g., “X *increase the risk of* Y”.

Compositional approach. Refers to methods for representing relations between two words that apply a compositional operator on their semantic representations rather than depending on relational patterns.

Unsupervised compositional operators. Following the literature, when a compositional operator does not have learnable parameters, we call it unsupervised.

Relational walk. A theoretical model that derives a relationship between the connections between entities and relations in a knowledge graph and the corresponding entity and relation embeddings. The relational walk is a generative model that performs a random walk over the KG parametrised by hidden knowledge vectors.

Relation compositions in knowledge graphs. The task of predicting representations for novel relation types by composing pre-trained relation embeddings for the relations that exist in a knowledge graph.

1 Introduction

1.1 Context of the Study

> *"Every word in a sentence is not isolated as it is in the dictionary. The mind perceives connections between a word and its neighbours. The totality of these connections forms the scaffold of the sentence."*
>
> - Tesniére, 1959

In natural languages, i.e., English, Arabic, Chinese, relations are the connections between concepts that are expressed by words[1]. As stated by cognitive scientists, understanding words by their relationships is essential to the expression of continuous thoughts for human intelligence (Tesnière, 1959). In aptitude tests provided by government institutions to enter universities in some countries such as the United States[2] and Saudi Arabia[3], students are asked to detect word analogies of the form $a : b :: c : d$ (read as a is to b as c is to d) to measure their ability to recognise relations. One has to identify and then compare the relationship between the two words in each pair (a, b) and (c, d) to answer these type of questions. For example, (*lion*, *cat*) is relationally similar (i.e., analogous) to (*ostrich*, *bird*) because a *lion* is a large *cat* as an *ostrich* is a large *bird*.

Nowadays, in the digitalised world, it is required for a computational system that processes human languages to be equipped to learn semantics not only for the words but also for relations between them to intelligently perform useful tasks just as humans do. Such tasks that benefit from using relational information within Natural Language Processing (NLP) include analogical reasoning (Turney and Littman, 2005; Li et al., 2018),

[1] Throughout this thesis, words may refer to unigrams (e.g., *food* and *animal*) as well as multi-word expressions including named entities such as person or location names (e.g., *United States*). The terms word and entity are used interchangeably in this thesis.

[2] `https://collegereadiness.collegeboard.org/sat`

[3] `https://qiyas.sa/en/Exams/Education/GeneralAbilities/Pages/default.aspx`

relation extraction (Mintz et al., 2009; Su et al., 2018), relational search (Cafarella et al., 2006; Duc et al., 2010, 2011), machine translation (Nakov, 2008; Zhang et al., 2018), text categorisation (Espinosa-Anke and Schockaert, 2018; Camacho-Collados et al., 2019), textual entailment (Vu and Shwartz, 2018; Joshi et al., 2019), word-sense disambiguation (Federici et al., 1997; Agirre et al., 2014) and knowledge graphs completion (Nguyen, 2017), among others.

In relational search, given a query "*a* is to *b* as *c* to ?" the goal is to retrieve entities that have a semantic relationship with *c* similar to that between *a* and *b*. For example, given the relational search query "*Bill Gates* is to *Microsoft* as *Steve Jobs* is to ?", a relational search engine is expected to return the result *Apple Inc.* because of the CEO relation that holds between the first and the second entity pairs. One might also want to ask a search engine to list all entities that are in a relation with another entity. For instance, a query "*what can be produced from a cork tree*", needs a relational search to retrieve entities that are in the produced-from relation with *cork tree*. Recognising textual entailment offers another kind of application. Given a premise P, "*a man ate an apple*", and a hypothesis H, "*a man ate a fruit*", a model that can infer the existence of the is-a relation between *fruit* and *apple* would correctly predict that H entails P.

Modelling the meaning of relations and relational reasoning are well-defined problems in the field of NLP and machine learning. Resources that are adopted for such tasks can be categorised into either: (a) text-based, and (b) knowledge graphs-based. Text corpora play a critical role in NLP research. In the context of learning relations, linguistic contexts of entities in a corpus work in one way or another as clues to infer relations between entities. For example, the sentence from Wikipedia that says "***Paris** is the capital and most populous city of* ***France**, with ...*" shows the capital-of relation. On the other hand, real-world facts such as *Paris* is the capital city of *France* is asserted in structured knowledge graphs such as Freebase, wherein *Paris* and *France* are represented by two nodes and the relation capital-of is the label for the edge that connects the two entities.

In NLP systems, linguistic components such as words, relations, phrases, etc. have to be represented in a way so as to make it possible for a computer to understand the underlying semantics. To address this need, Vector Space Models (VSMs) of semantics have been applied successfully as testified in the voluminous NLP related literature. The underlying concept of the VSMs is that a linguistic item is mapped to a vector of multiple dimensions corresponding to features that collectively derive the meaning of the represented item. These features are extracted and learnt from various resources such as the above-mentioned ones, text corpora and knowledge graphs. In theory, VSMs require semantically similar items to be nearby in space. The VSMs initially show their potential applications in document retrieval, where a search engine retrieves relevant documents to a given query according to the similarities (e.g., cosine of angles) between the query vector and the document vectors in a space (Manning et al., 2008). Documents and queries are represented in a multi-dimensional space where each term in the vocabulary is a dimension. In the context of word representations, in which VSMs have achieved breakthrough performance, the dimensions in a space basically indicate the co-occurrence statistics from a text corpus (discussed in details in Section 2.5.1).

Semantic spaces have been successfully extended to encode relations, word-pairs and to represent Knowledge Graphs (KGs) as well (Turney and Pantel, 2010; Bordes et al., 2013). Word-pairs are represented in vector space such that pairs of words that belong to the same relation are closer than word-pairs from different relation types. Text-based resources have been employed in different themes to obtain representations for word-pairs in space. One methodology heavily depends on particular contexts in which related words appear in a text corpus (Turney, 2005; Espinosa-Anke and Schockaert, 2018). In contrast, from 2013 onwards, word representations that are generated from text corpora using deep learning methods have shown promising performance in capturing relational features of word-pairs. This thesis investigates such a property in the VSMs of semantics for words towards relations. In the context of KGs, entities and relations are also represented in a semantic space such that entities that participate in similar relations are embedded closely to each other, while at the same time relations that hold between similar entities are embedded closely to each other in the relational embedding space. This thesis also studies representations of relations in KGs.

Having introduced the importance and resources of relational learning, **the work presented in this thesis is directed at several research issues within the context of representing relations between words**. The rest of this chapter is organised as follows. Section 1.2 introduces the research aim and motivations. In Section 1.3 the research issues that are considered for this thesis are listed. The contributions of this thesis are explained in Section 1.4. Published work and the thesis outline are presented respectively in Section 1.5 and Section 1.6.

1.2 Research Aim and Motivations

How to accurately represent semantic relations between words or entities remains a problem that is underexplored within the NLP community. As elaborated earlier, a text corpus represents an important and abundantly available source of information for building NLP systems. Traditionally, automatic learning of word pair representations from a text corpus has relied on *distributional relation analysis*, which states that the relation between two words in a pair is evidenced by the contexts that co-occur with the pair (Turney and Littman, 2005; Turney, 2006; Bollegala et al., 2008). Such contextual information is expressed in the form of lexico-syntactic patterns, for instance, "X *is a* Y" and "Y *such as* X" patterns indicate that X is a hyponym of Y (Snow et al., 2005). This approach for representing

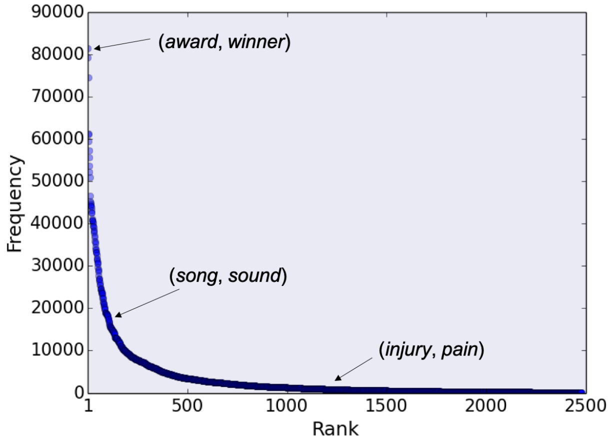


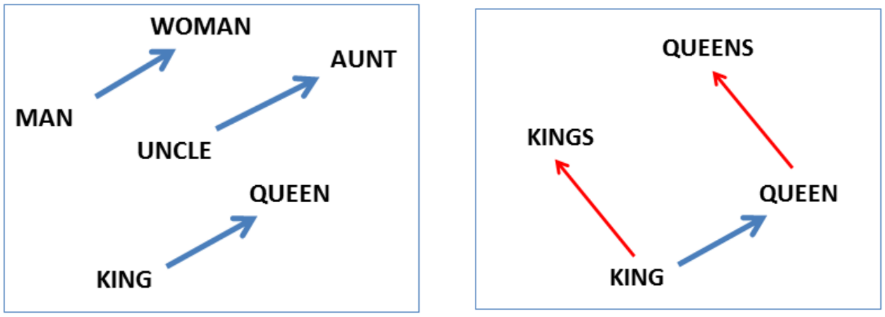
Figure 1.1: The frequency of related word-pairs in Wikipedia corpus.

relations is called, in this thesis, *pattern-based* because of the need for lexical patterns that connect pairs of words to infer relations between words.

Despite the good performance of the pattern-based approach in previous research (Turney, 2005; Jameel et al., 2018), it suffers from data sparseness which in turn limits the generalisation capabilities of this methodology. Specifically, two words have to co-occur in a contextual window, or else no relation can be represented for unseen word-pairs. Even in a large text corpus, not every related pair co-occurs within a specified window. Also, in any text collection, the number of sentences in which the two related words appear might be quite small or might not be relevant for characterising the considered relations, which drastically affects the quality of the relation representation. For example, Figure 1.1 shows the co-occurrence frequency and rank of related word-pairs[4] in the Wikipedia corpus. The graph illustrates that even in a large corpus as in the case of Wikipedia, the distribution of co-occurrences of related word-pairs has a long tail. Moreover, 321 word-pairs from the set of related words do not co-occur within any sentence.

To address the aforementioned sparsity problem of the pattern-based approach for relation representations, we have to obviate the strong co-occurring assumption of related word-pairs to represent relations. Mikolov et al. (2013c) showed that pre-trained word representations (a.k.a embeddings), which are low-dimensional real-valued vectors that are generated typically from distributional information in text, encode remarkable structural properties about semantic relations. As illustrated in Figure 1.2a, the vector offsets $\boldsymbol{woman} - \boldsymbol{man}$, $\boldsymbol{queen} - \boldsymbol{king}$ and $\boldsymbol{aunt} - \boldsymbol{uncle}$ are approximately parallel, here the notation $\boldsymbol{man}$ is used to denote the embedding of the word *man*. These vector offsets describe *gender*

[4] These word-pairs are taken from two popularly used benchmarks for relations, namely, SAT and SemEval-2012 Task 2 that are introduced later in Section 3.3.1 and Section 3.3.2, respectively.



(a) Gender relation

(b) Plural relation

Figure 1.2: Example of linguistic regularities in word embeddings. The figure is taken from Mikolov et al. (2013c).

directions from male to female in the embedding space. The well-known example is that of the proportional analogy expressed in (1.1). Mikolov et al. (2013c) fill the blank of the analogy by finding the nearest neighbour word vector to the composed vector $\boldsymbol{king} - \boldsymbol{man} + \boldsymbol{woman}$.

$$man : woman :: king :? \quad \text{(semantic)} \tag{1.1}$$

$$king : kings :: queen :? \quad \text{(syntactic)} \tag{1.2}$$

Such an approach is referred to as *compositional* because (a) the way in which the relation representation is composed by applying some algebraic relational operator on the representations of the words that participate in a relation, and (b) we infer a relation between a pair without assuming the availability of patterns in which a pair matches in a corpus. In this thesis, compositional methods for relation representations are explored from multiple aspects as will be demonstrated in the next section.

As elaborated in the previous section, unstructured text corpora are not the only source for relations. KGs represent another important source for relational information that organise knowledge bases of relations in the form of nodes (entities) and edges (relations). While relations in text-based sources are latent as they are induced through linguistic patterns, KGs typically consist of a set of well-defined discrete relation types. Such KGs are either constructed manually or using relation extraction techniques on a text corpus (Mintz et al., 2009; Riedel et al., 2010). The advent of KGs in different domains is important for a wide range of NLP tasks such as learning knowledge-enhanced semantics (Faruqui et al., 2015a; Alsuhaibani et al., 2018), question answering (Das et al., 2017) and text summarisation (Baralis et al., 2013). However, the constructed KGs, especially those extracted from text, suffer from sparsity since many plausible facts are missing (Pujara et al., 2017; Paulheim, 2017). One popular way to handle sparsity in KGs is to encode entities and relations of a KG into low-dimensional vector spaces, so-called Knowledge Graph

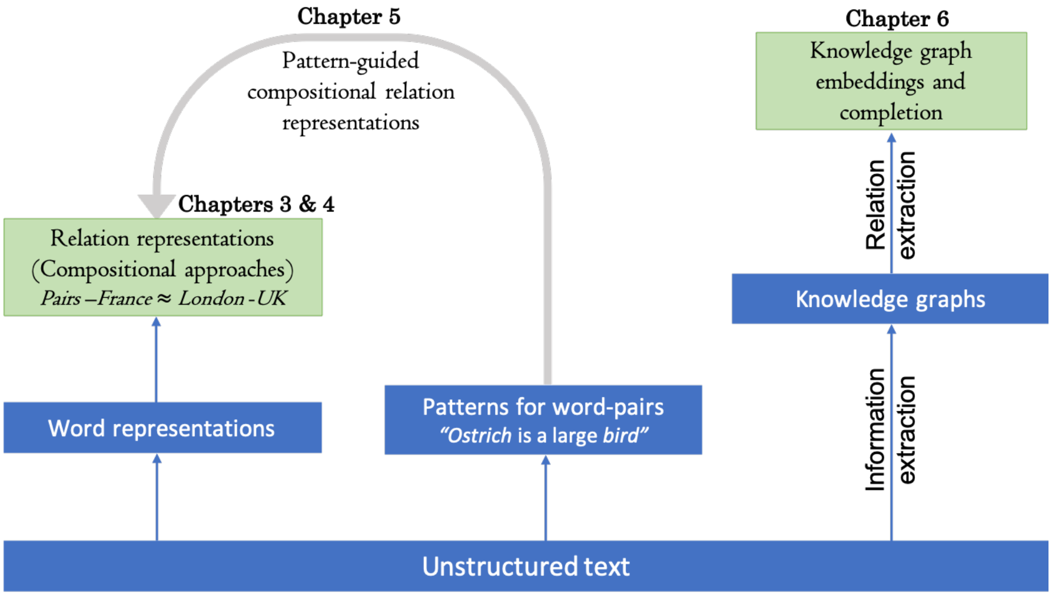

Chapter 5
Pattern-guided compositional relation representations
Chapter 6
Knowledge graph embeddings and completion
Chapters 3 & 4
Relation representations (Compositional approaches)
Pairs – France ≈ London - UK
Relation extraction
Knowledge graphs
Word representations
Patterns for word-pairs
"Ostrich is a large bird"
Information extraction
Unstructured text

Figure 1.3: An illustration of diverse sources that are consider in this thesis for learning relation representations.

Embeddings (KGEs). As is customary in the literature, entity and relation embeddings are jointly learnt such that some objective that models the interaction between two entities and a relation that holds between them is optimised (Bordes et al., 2013; Yang et al., 2015; Nickel et al., 2015; Guo et al., 2016; Nickel et al., 2016). By embedding entities and relations that exist in a KG in some space, we can infer previously unseen relations between entities, thereby expanding a given KG.

Although existing KGE methods demonstrated valuable performance in predicting new links between entities, objectives for learning such embeddings are heuristically motivated. Therefore, a theoretical understanding of KGE methods is required. In addition, KGE methods embed existing entities and relations and they are unable to predict representations for unseen relation types. Accordingly, this thesis contributes to solving the above-raised issues of KGEs.

Figure 1.3 shows the different sources of information that are considered in this thesis for relation representations. From an unstructured text corpus, word embeddings are learnt from the corpus and are taken as a source of information for learning compositional operators for word-pair representations. Such compositional methods can be improved by supplying patterns of related word-pairs during training (pattern-guided compositional relation representations). As shown on the right side of the figure, extracting relations from a KG in the form of knowledge graph embeddings to reduce sparsity in the KG is also considered in the conducted research. The section below will present the research question and related issues that are studied in this thesis.

1.3 Research Question and Issues

From the foregoing aims and motivations of this thesis, the main research question considered in this work is as follows: *Can we learn relation representations from word representations; and if so what are the appropriate methods and resources for achieving this?*

Many issues arise to answer the above question, here are the addressed issues:

- Given pre-trained word embeddings, what is the best unsupervised compositional operator to represent relations between words? and how appropriate is such an operator for various relation types?
- Can we discover discriminating relational features from word representations to measure the relational similarity between two word-pairs?
- Can we systematically investigate a bilinear operator, which is parametrised by a 3D tensor, to map two given word embeddings into a vector representing a relation between the two words?
- Can we learn better compositional operators for relation representations from word representations?
- Can we improve the performance of compositional relation representation methods by training such methods using the two sources of information, namely: (a) word-embeddings of related pairs, and (b) co-occurring patterns extracted from a corpus?
- Given a KG, can we enrich the graph by inferring missing links using a theoretically motivated approach for relation and entity embeddings?
- Given pre-trained KGEs, can we infer embeddings for unseen (i.e., novel) relations using existing relation embeddings and relational logical rules?

1.4 Contributions

The primary goal of this thesis is to explore resources and methodologies to represent relations between words such that we overcome data sparsity of related pairs in texts or KGs. The thesis makes a number of noteworthy contributions, which are:

- **A comparative study for unsupervised compositional operators to derive relational features from word embeddings.** The contribution of this study aims to compare different unsupervised compositional operators for obtaining relation representations from word-level representations. Such unsupervised operators do not have learnable parameters, and they are applied on word representations learnt in unsupervised settings. The performance of such unsupervised compositional methods are investigated by measuring the relational similarities using several relational benchmark datasets, and also evaluated in a KG completion task. The performance of the compositional operators among various relation types is also investigated. This contribution was published in the Knowledge-Based System journal (Hakami and Bollegala, 2017), and is presented in Section 3.4 of this thesis.

- **A method to discover discriminative features in word representations for measuring relational similarities.** Features that accurately express the relational similarity between two word-pairs remain largely unknown. So far, methods have been proposed based on linguistic intuitions such as the functional space proposed by Turney (2012), which consists purely of verbs. In contrast, a data-driven approach for discovering feature spaces for relational similarity measurement is proposed in this thesis. This study was published at PACLING 2017 (Hakami et al., 2017). Section 3.5 is devoted to this contribution.

- **A mathematical analysis for bilinear relation representations.** A simple, yet surprisingly accurate method for representing a relation between two words is to compute the vector offset (PairDiff) between their corresponding word embeddings. Despite the empirical success, it remains unclear as to whether PairDiff is the best operator for obtaining a relational representation from word embeddings. To this end, a theoretical analysis is conducted of generalised bilinear operators that can be used to measure the ℓ_2 relational distance between two word-pairs. This work was published at COLING 2018 (Hakami et al., 2018), and is included in Chapter 4 of this thesis.

- **A method to learn compositional operators for relation representations.** Despite that simple unsupervised operators such as the vector offset between two-word embeddings have shown to recover certain relationships between words, how to accurately learn generic relation representations from word representations remains unclear. In this thesis, relation representation is modelled as a supervised learning problem and parametrised operators are learnt such that they map pre-trained word embeddings to relation representations. Specifically, a method for learning relation representations using a feed-forward neural network that performs relation prediction is proposed. This contribution was published at Automated Knowledge Base Construction (AKBC) 2019 (Hakami and Bollegala, 2019b), and is presented in Section 5.2 of the thesis.

- **An exploration of the complementary of lexical patterns and word embeddings for representing relations.** A semantic relation between two given words can be represented using two complementary sources of information: (a) the semantic representations of the two words and, (b) the lexico-syntactic patterns obtained from the co-occurrence contexts of the two words. Pattern-based approaches suffer from sparsity, while methods that rely only on word embeddings for the related pairs lack relational information. To this end, a Context-Guided Relation Embedding model that uses the two sources of information is proposed under a self-supervised fashion. The learnt operator is evaluated for its ability to create relation representations for word-pairs that do not co-occur. The code and pre-trained word-pair embeddings are publicly available for reproducibility[5]. The proposed method was published at PACLING 2019 (Hakami and Bollegala, 2019a), and is presented in Section 5.3.

- **A proposed relational walk model for KGEs.** Although existing KGE methods demonstrate good empirical performance on predicting missing links in a graph, theoretical understanding of KGE methods is comparatively underdeveloped. As such, it is not clear how the heuristically defined KGE objectives relate to the generative process of a KG. This thesis seeks to fill this void by providing a theoretical analysis of KGEs. To do so, the random walk model of Arora et al. (2016) of word embeddings was extended in this thesis to KGEs to derive a scoring function that evaluates the strength of a relation between two entities. Specifically, a generative process is proposed where the formation of a relation between two entities using the corresponding relation and entity embeddings is derived theoretically. This contribution is presented in Section 6.2 of this thesis.

- **A proposed compositional semantic method to infer embeddings for novel relations.** Existing KGE methods can only learn representations for the relations that exist in a KG. However, there are correlations among relations in a graph that allow relation inferences. For instance, given that X born-in Y and Y capital-of Z, we can infer the relation X nationality Z. *Relation composition* is introduced in this thesis as the task of inferring embeddings for unseen relations by combining existing relations in a KG. Specifically, a supervised method (modelled as a non-linear neural network) is proposed to compose relational embeddings for novel relations using pre-trained relation embeddings for existing relations. The implementation of the proposed relation composition model is publicly available on GitHub[6]. The work was published at PACLING 2019 (Chen et al., 2019), and included in Section 6.3

1.5 Publications

Most of the material presented in the thesis has already been published in NLP conferences and journals. The list of the publications (chronological order) and their corresponding

[5] https://github.com/Huda-Hakami/Context-Guided-Relation-Embeddings

[6] `https://github.com/Huda-Hakami/Relation-Composition-for-Knowledge-Graphs`

chapters or sections is presented below:

1. Danushka Bollegala, Huda Hakami, Yuichi Yoshida, Ken-ichi Kawarabayashi: *RelWalk - A Latent Variable Model Approach to Knowledge Base Embedding*, under review. **Section 6.2**.

2. Huda Hakami and Danushka Bollegala: *Context-guided Self-Supervised Relation Embeddings*, Proc. of the 16th International Conference of the Pacific Association for Computational Linguistics (PACLING), October, 2019. **Section 5.3**.

3. Wenye Chen, Huda Hakami and Danushka Bollegala: *Learning to Compose Relational Embeddings in Knowledge Graphs*, Proc. of the 16th International Conference of the Pacific Association for Computational Linguistics (PACLING), October, 2019. **Section 6.3**.

4. Huda Hakami and Danushka Bollegala: *Learning Relation Representations from Word Representations*, Proc. of the Automatic Knowledge Base Construction Conference (AKBC), May, 2019. **Section 5.2**.

5. Huda Hakami, Kohei Hayashi and Danushka Bollegala: *Why does PairDiff work? - A Mathematical Analysis of Bilinear Relational Compositional Operators for Analogy Detection*, Proc. of the 27th International Conference on Computational Linguistics (COLING), pp. 2493-2504, 2018. **Chapter 4**.

6. Huda Hakami and Danushka Bollegala: *Compositional approaches for Representing Relations Between Words: A Comparative Study*, Knowledge-Based Systems (KBS), Vol. 136, pp. 172-182, 2017. **Section 3.4**.

7. Huda Hakami, Angrosh Mandya, and Danushka Bollegala: *Discovering Representative Space for Relational Similarity Measurement*, Proc. of the 15th International Conference of the Pacific Association for Computational Linguistics (PACLING), pp. 76-87, 2017. **Section 3.5**.

1.6 Thesis Outline

The overall structure of the reminder of this thesis takes the form of five main chapters and a concluding chapter as follows.

Chapter 2: Background and Related Works. This chapter consists of background knowledge about relations between words that is important to understand the remainder of this thesis. The chapter introduces semantic spaces in the NLP field. The applications of relational reasoning in NLP are demonstrated. Different methods in the literature for representing relational information with respect to the work presented in this thesis are also reviewed in this chapter.

Chapter 3: Deriving Relational Features from Word Representations. The first part of this chapter introduces the word embedding models and relational benchmark datasets that are used extensively in this thesis. Then, the chapter presents a systematic comparison of four different unsupervised compositional operators for representing relations, namely: offset, concatenation, addition and multiplication. The last part of the chapter presents a proposed data-driven approach to discover discriminative features form word-level representations for measuring relational similarity between word-pairs.

Chapter 4: Mathematical Analysis for Bilinear Relation Representations. A theoretical analysis of generalised bilinear operators that can be used to measure the ℓ_2 relational distance between two word-pairs is presented in this chapter. Specifically, a theorem of a bilinear relational operator is proved. The chapter includes empirical validations of the theory to show the rationality of the presented analysis.

Chapter 5: Learning Compositional Operators for Relation Representations. This chapter presents learnable compositional operators to represent relations between words. The first section is about the proposed supervised operators for relation embeddings using word embeddings. The second section introduces self-supervised context-guided compositional relation embeddings, which explore the complementarities of training on both word-embeddings and relational patterns.

Chapter 6: Relations in Knowledge Graphs. This chapter considers the problem of representing relational information in a KG for KG completion. The first section presents a theoretically analysed relational walk model for KGEs. The second section shows a relation composition, which is the task of predicting relation embeddings for novel relation types by composing the embeddings for existing relation types.

Chapter 7: Conclusion. This chapter summarises the main findings of this thesis and discuss potential future directions.

1.7 Summary

This chapter has introduced an overview of the research area considered in this thesis. The research aim, motivations and questions were defined. A summary of the main contributions alongside the published articles was also presented. The chapter was concluded with an outline of the structure of the thesis. The next chapter provides background information and a review of the related literature required to understanding the remainder of the thesis.

2

Background and Related Work

2.1 Introduction

As elaborated in Chapter 1, this thesis is concerned with representing relations between words in texts or KGs. This chapter will provide a review of the background information and the related literature on representing relations in the context of NLP. The chapter begins with a brief overview of semantic representations for natural language components in Section 2.2. Then, Section 2.3, provides motivational answers to the question as to why we should care about relations between words. This is followed by Section 2.4 which presents the classical pattern-based approach for representing relations between words in texts. After presenting the pattern-based approach along with its drawbacks, Section 2.5 introduces an alternative approach to capture relational information in a word-pair from the representations of corresponding words in the pair. A hybrid approach that combines the two resources (i.e., patterns and word representations) is presented in Section 2.6. Section 2.7 then introduces multi-relational KGs and related work concerning the task of embedding relations using KGE methods.

2.2 Semantic Spaces in NLP

A well-defined field of linguistics that concerns the analysis of meanings is called *semantics*, with *lexical semantics* being a sub-field that deals with meanings of individual words and relations between them (Cruse et al., 1986). We can describe the meaning of a word by considering the word as a container (semantic features) or through its relationships with other words (lexical relations). For example, *colour* can be a feature when representing the word *flower*. Considering lexical relations, we can explain the meaning of the word *daffodil* in terms of its relationship to *flower* because "*daffodil* is a kind of *flower*".

As humans, we use natural languages to communicate and understand the world around us. With the increasing amounts of textual data, it is important to digitise this field and enable computers to understand languages that we humans speak and write. To this end, NLP researchers seek to propose various methods to represent the semantics of linguistic items in such a way that computers can process, reason and perform useful tasks on texts. Semantic VSMs is the most dominant research area in the field of computational semantics. Generally speaking, VSMs make use of geometric spaces to assign data points to vectors using a set of features, where each feature is considered to be a separate dimension in a vector space with a real value. Selecting such features to construct a semantic space profoundly depends on the nature of a concept whose meaning we aim to encode. The key aspect in VSMs is that similar concepts are represented closely in a space. In contrast to discrete representations, the notion of similarity can be directly inferred from VSMs by measuring degrees of similarities between the representations of the corresponding concepts.

Turney and Pantel (2010) provide a comprehensive literature survey about VSMs of semantics that are proposed to represent various linguistic concepts such as words, word-pairs, phrases and documents. Typically, VSMs are constructed automatically relying on the distribution of the concepts to be represented in a text corpus. Impressive successes has been shown for VSMs in information retrieval, analogical reasoning and KG completion, among others. Take for instance the task of analogical reasoning, word-pairs can be represented in a multi-dimensional space where, for instance, linguistic contexts that connect word-pairs are the dimensions of the space, and the elements correspond to the number of times a particular pair appears with a given context. Such representations can be used to measure the relational similarities between word-pairs to find analogies or can be used as features to downstream NLP applications. For KGEs, each relation can be represented as a latent vector that works as an operator on entity vectors to predict relations between entities.

The work in this thesis is broadly connected to three different types of VSMs: (a) word meanings, (b) word-pair representations and (c) KGEs. This chapter presents the necessary background concerning each of these semantic spaces with respect to the conducted research in the thesis. The main focus in this thesis is directed at encoding relational information between words in a semantic space from different aspects as elaborated previously in Sections 1.3 and 1.4.

2.3 Why Do We Care About Relations in NLP?

A relation is a way of describing how two things are connected. Connections exist everywhere around us, and they vary across different domains, ranging from relationships between people that represent a business, to social relations, to mathematical relations between quantities such as equivalence and reflexive relations. In the field of NLP, relations mainly refer to semantic links that hold between words and can be used to define the nature of word meanings in a language clearly. In this section, we will introduce some properties of relations between words (Section 2.3.1). Then, in Section 2.3.2 we will show how representing relations between words plays a vital role in numerous NLP tasks.

2.3.1 Relations Between Words

In a written or spoken language, a word is the basic lexical unit that is used with others to create a meaningful phrase or sentence. Various types of relations exist between words such as the Hypernym between *ostrich* and *bird*, the Antonym between *hot* and *cold*, the Meronym between *car* and *engine* and the Attribute between *glass* and *fragile*. If we consider named entities, we can observe a richer diversity of relations such as Founder-of between *Bill Gates* and *Microsoft*, Capital-of between *Tokyo* and *Japan* and CEO between *Tim Cook* and *Apple Inc.* Extensive efforts have been made in the literature by linguistics, cognition science and computational linguistics to define taxonomies of relationships between words (Casagrande and Hale, 1967; Levi, 1978; Chaffin and Herrmann, 1984; Nastase and Szpakowicz, 2003; Hendrickx et al., 2009; Jurgens et al., 2012). Many of the datasets used are publicly available for researchers.

Relations between words exist in many flavours. Some relations exist between two words relying on a given specific context in which the two words co-occur. For example, given the sentence "the *machine* makes a lot of *noise*", we can infer a Cause-Effect relation between *machine* and *noise* considering the linguistic clue *makes a lot of* in this particular sentence. The former interactions that are sensitive to contexts are referred to as *syntagmatic* relations, a phrase first coined by Saussure (1959) and then adopted by other studies about relations (Khoo and Na, 2006; Hendrickx et al., 2009). For such syntagmatic relations, we might infer different relations between the same word-pair based on the provided contexts. On the other hand, Saussure (1959) defines *associative* (paradigmatic) relations as those that hold between two words regardless of the contexts in which they co-occur. For instance, the capital-of relation between *London* and *England* and is-a relation between *cat* and *animal*. Associative relations can be lexical (is-a, part-of) or encyclopaedic (capital-of, nationality). Prior work has also considered associative syntactic relations between words such as plural (*cat*, *cats*) and comparative (*weak*, *weaker*) depending on word morphology (Mikolov et al., 2013c; Vylomova et al., 2016). There are also implicit relations within a noun compound, which is a sequence of two nouns acting as a single noun in English (Downing, 1977). Examples of implicit relations in noun-compounds are cause in *flu virus*, container in *apple cake* and location in *home town*. Several researchers have attempted to automatically recognise implicit relations among noun compounds (Nastase and Szpakowicz, 2003; Tratz and Hovy, 2010; Shwartz and Dagan, 2018).

Another dimension of variation across relations is the number of linked arguments. In this sense, a binary relation is a link that exists between two arguments. However, relations might connect more than two arguments, which are termed n-ary relations (where n indicates the number of linked arguments). For example, the ternary response relation in the instance (*EGFR*, *L858E*, *gefitinib*) means that the *EGFR* mutation in the gene *L858E* responds to the drug *gefitinib*. The interest in this thesis is directed in particular at binary relation types; the extension to n-ary relations can be investigated in future work.

2.3.2 Applications of Relation Learning

Reasoning about relations between entities, rather than about individual entities, is an essential element of human cognition (Glass et al., 1977; Penn et al., 2008). For computational systems, connecting up entities within a chunk of text performs a battery of NLP tasks in which relational information is essential to conduct the tasks successfully. Such tasks include recognising word analogies, relational information retrieval, textual entailment, machine translation, metaphor detection and knowledge base completion.

Relation representations have been successfully used for relational similarity-based tasks ever since the work of Turney (2005), who demonstrated that the similarity between two pairs could be measured using their relation representations. In the Scholastic Aptitude Test (SAT), a standardised test widely used for college admissions in the United States, an analogy question consists of a stem word-pair and a list of four to five pairs in which only one represents the correct choice for the stem. To answer analogical SAT questions, one is required to determine relations between the pairs to be compared. Turney (2005) proposed an algorithm for representing relations that achieved human-level performance as the average of US college applicants is 57.0%, whereas the state-of-the-art algorithm gave 56.1% accuracy.

In areas of research involving cross-sentence inference, such as textual entailment and question answering, incorporating relational representations between words is beneficial. For textual entailment tasks, given a premise P "*a man ate an apple*" and a hypothesis H "*a man ate a fruit*", a model that can infer the existence of is-a relation between *fruit* and *apple* would correctly predict that P entails H. A recent study by Joshi et al. (2019) shows the importance of enriching word representations with relation representations to improve cross-sentence inferences.

Relational reasoning has also been applied to the task of machine translation (Nakov, 2008; Aharoni and Goldberg, 2017; Zhang et al., 2018). For example, Nakov (2008) paraphrases implicit relations in noun compounds of sentences in a source language to generate new sentence variants that are combined with training data. Further, Zhang et al. (2018) improve recurrent neural network encoder-decoder models for translations by learning pairwise relations between words in a source sentence while decoding to a target sentence.

Metaphorical language is ubiquitous in our life, which is typically the use of a word to an object to which it is not literally applicable such as "*The printer died*", here *died* is a

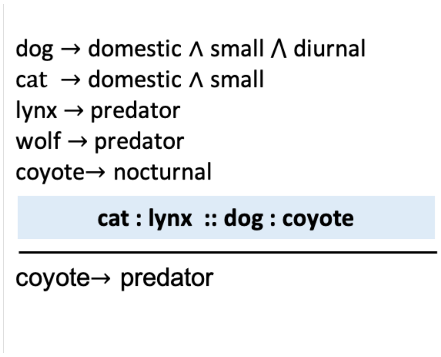


Figure 2.1: Example of analogical reasoning for completing rule bases. Taken from Schockaert and Prade (2014).

metaphorically used verb to describe that the *printer* cannot be restarted (Lakoff and Johnson, 2008). Automatic detection of metaphorical and literal language in discourse has been extensively studied (Barnden and Lee, 2001; Zayed et al., 2018). Interestingly, metaphors are similar to analogies since both employ comparisons between two concepts (Gentner et al., 2001). As such, an implicit comparison in a metaphor can be expressed as an explicit analogy, e.g., "*The printer died*" expressed as *person* : *die* :: *printer* : *damage* (Turney, 2006). Consequently, reasoning about relations between words has also its potential applications in identifying metaphors.

Automatic KG completion is another important area of research in which relation representations are successfully applied. Numerous studies have shown that representing entities and relations within a given KG enables predicting missing facts and thus resolves the poor coverage of such graphs (Socher et al., 2013a; Nguyen et al., 2016; Dettmers et al., 2018; Bollegala et al., 2019). Similarly, to deal with missing domain knowledge in rule-based systems, we can use the assumption that analogous changes in the condition of a rule lead to equivalent changes in the conclusion. Figure 2.1 illustrates an example of predicting the missing plausible rule ($coyote \rightarrow predator$) from a proportional analogy *cat* : *lynx* :: *dog* : *coyote* and a given set of if-then rules.

Having discussed the potential applications of relation learning, we now move on to present the most popular approach for representing relations between words, namely the pattern-based approach.

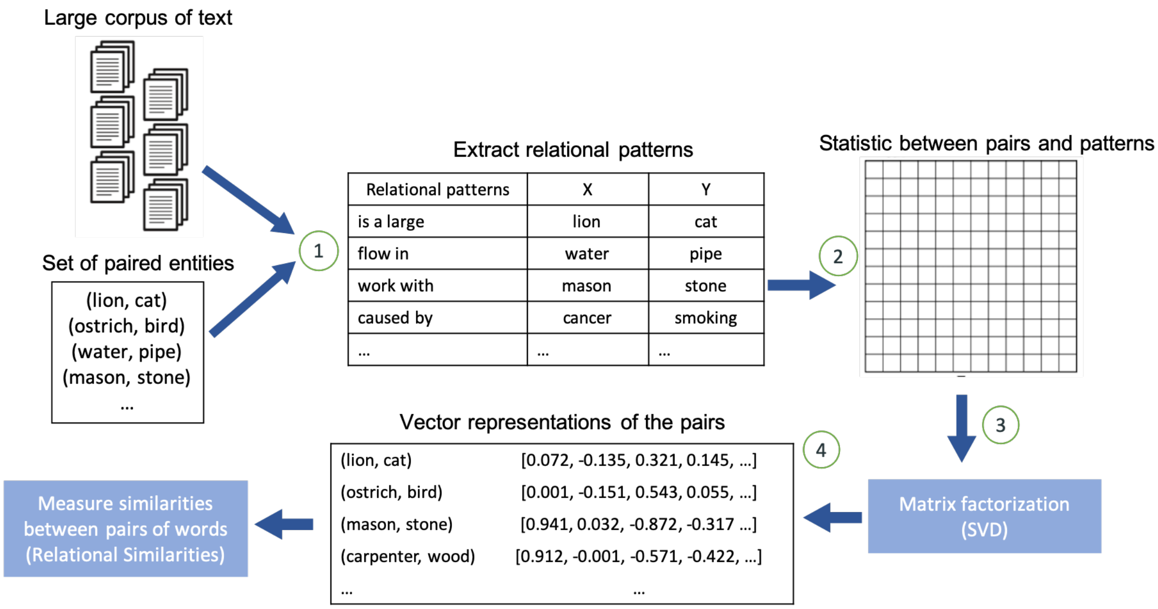
2.4 Pattern-Based Approach for Relations

Several methods have been proposed for learning representations that encode the relationship between two words. This section will introduce the most popular pattern-based strategy for relations, which essentially relies on the contextual patterns that link pairs of words in a text corpus. A definition of linguistic patterns in the context of relational information, with examples, is discussed in Section 2.4.1. Next, Latent Relational Analysis is presented, which is a pioneering pattern-based model for relations, followed by other recent variants of pattern-based relation representation methods. Section 2.4.4 concludes this part of the chapter by discussing the limitations of the pattern-based approach, which in turn motivate the work presented later in this thesis.

2.4.1 Relational Patterns

An unstructured text corpus forms an important resource to extract information for numerous NLP tasks such as relation extraction where the task is to identify the relation that holds between two named entities (Hearst, 1992; Mintz et al., 2009; Baldini Soares et al., 2019). The linguistic contexts in which two words co-occur in the corpus provide useful clues regarding the relations that exist between the two related words. Such contexts that connect two related entities are expressed as relational patterns. For example, the causality relation between *smoking* and *lung cancer* can be expressed by multiple text spans as follows: "*smoking* increases the risk of *lung cancer*", "*smoking* led to *lung cancer*" and "*smoking* causes *lung cancer*". Replacing the targeted entities with placeholders produces general patterns that can be matched with any other pairs of entities. For instance, "X *increase the risk of* Y" is a pattern that might match word-pairs such as (*smoking*, *lung cancer*), (*obesity*, *diabetes*) and (*explosion*, *damage*). Here, X and Y in a pattern refer to the first and the second entity of a pair, respectively.

Several studies have defined hand-crafted patterns for lexical relations. Hearst patterns are well-known manually constructed lexico-syntactic patterns for the Hypernym taxonomic relation (Hearst, 1992). Hearst defines a small set of patterns for hypernym detection; for example, "X *and other* Y", "*Such* Y *as* X" and "X *is a* Y *that*". Lexical patterns for other relations have also been defined such as Meronym (Berland and Charniak, 1999; Girju et al., 2003), causality (Marshman, 2002) and protein-protein inhibit relations (Pustejovsky et al., 2001). Later on, the limited pre-defined set of patterns for relations was replaced by automatic extraction of lexico-syntactic patterns from a text corpus (Snow et al., 2005; Turney, 2005; Shwartz et al., 2016; Washio and Kato, 2018b; Joshi et al., 2019). Given a set of entity pairs, the sentences containing the two words in each pair are extracted to generate the patterns. To increase the precision of the set of patterns, sentences in which the two words are far apart from each other are not considered because the assumption is that the longer the distance, the less useful the pattern for the relation. Many features can be extracted from the patterns—the two popular types are lexico-syntactic surface tokens (Hearst, 1992; Turney et al., 2003) and dependency paths that give dependencies between tokens in a pattern (Yangarber et al., 2000; Nakashole et al., 2012; Shwartz et al., 2016).


Large corpus of text
Set of paired entities
(lion, cat)
(ostrich, bird)
(water, pipe)
(mason, stone)
...
1
Extract relational patterns
2
Statistic between pairs and patterns
3
Matrix factorization (SVD)
4
Vector representations of the pairs
Measure similarities between pairs of words (Relational Similarities)

Relational patterns	X	Y
is a large	lion	cat
flow in	water	pipe
work with	mason	stone
caused by	cancer	smoking
...	...	...

(lion, cat)	[0.072, -0.135, 0.321, 0.145, ...]
(ostrich, bird)	[0.001, -0.151, 0.543, 0.055, ...]
(mason, stone)	[0.941, 0.032, -0.872, -0.317 ...]
(carpenter, wood)	[0.912, -0.001, -0.571, -0.422, ...]
...	...

Figure 2.2: An illustration of LRA method for word-pair representations. This Figure is inspired by Figure 1 in Liu et al. (2017).

2.4.2 Latent Relational Analysis

Latent Relational Analysis (LRA) is a pattern-based method for representing word-pairs by adopting *latent relational hypothesis*, which was coined by Turney et al. (2003) and literally states that: "*pairs of words that co-occur in similar patterns tend to have similar semantic relations*". Figure 2.2 summarises the LRA steps for generating word-pair representations. The first step in the process of LRA is to align a given set of pairs with a corpus to extract relational contexts. Then, following the VSM of semantics (Turney and Pantel, 2010), each pair of words is represented using a vector of pattern frequencies, with the elements corresponding to the number of different sentences where the two words in a given pair co-occur with a particular pattern.

A two-dimensional visualisation of word-pair representations using the LRA method can be shown as indicated in Figure 2.3. We notice that the angle between (*ostrich*, *bird*) and (*lion*, *cat*) is much smaller than the angle between either of the pairs and (*mason*, *stone*). Because the number of automatically defined patterns in the pair-pattern matrix is large, a dimensionality reduction method, such as Singular Value Decomposition (SVD), is applied to obtain dense and latent representations for word-pairs. This representation allows us to measure the relational similarity between two given pairs of words by the cosine of the angle between the corresponding pattern-frequency vectors.

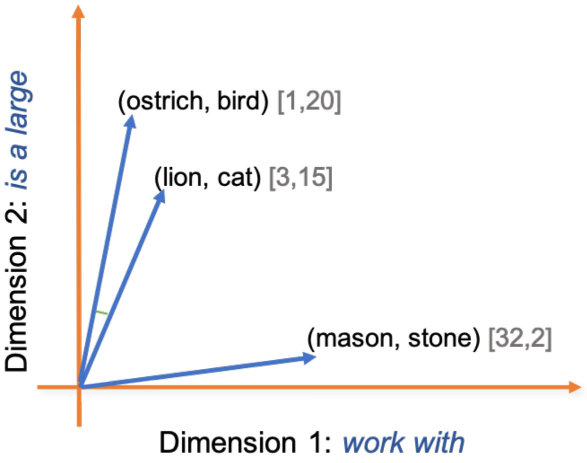


Figure 2.3: Illustrative visualisation of word-pair vectors. Here axes *work with* and *is a large* are the relational patterns. Three targeted word-pairs are plotted along with their statistical vectors that indicate co-occurrence frequencies between pairs and patterns.

2.4.3 Other Pattern-Based Methods

Proposed pattern-based methods differ with respect to how they encode pattern features in relation representations. As seen in LRA, each lexico-syntactic pattern is considered to be a dimension in the pair-pattern vector space that involves counting statistical co-occurrences for feature values. In recent years, with the emergence of deep learning, the focus in pattern-based relation representations has shifted to the use of neural networks. Hashimoto et al. (2015) and Fan et al. (2015) represent relations by averaging word embeddings for the words that occur in between a and b (more details on word embedding models is presented in Section 2.5.1). Along similar lines, in a semantic vector network model (SeVeN) that is proposed by Espinosa-Anke and Schockaert (2018), a relation vector for a word-pair (a, b) is defined as the average of the word vectors of the corresponding words in the surface patterns that match a and b augmented with a and b word vectors. On the other hand, Jameel et al. (2018) learn global relation vectors by optimising an objective function that can approximate the 3-way co-occurrence statistics between the word pair (a, b) and each context word that occurs with the pair in a sentence. Rossiello et al. (2019) learn relation representations of entity pairs by first aligning pairs in a KG to a corpus to extract sentences in which the pairs co-occur, then learning pair embeddings by predicting the analogy between entity-pair sentences.

The above-mentioned methods, along with LRA, require word-pairs to co-occur frequently in a corpus, which is a strong condition that suffers from the limitations explained in the following section.

2.4.4 Limitations of Pattern-Based Semantic Spaces

The majority of the previously discussed pattern-based methods for relation representations achieve successful performance on different relation-specific tasks. For instance, the LRA model achieved human-level performance for measuring relational similarity on the SAT multiple-choice word analogy questions. The global relation vectors proposed by Jameel et al. (2018) record the best result in relation induction, which is a binary classification task to decide whether a test pair is related in the same way as the pairs in a given set of word-pairs, compared to other relation representation methods. SeVeN also reports an increase in the performance of text categorisation and sentiment analysis tasks when individual word representations are augmented with their proposed relation vectors.

Although the use of relational patterns has shown good performance when it comes to representing the semantic relations between two words, this approach suffers from data **sparseness**. In the pair-pattern matrix, most of the elements have zero occurrences because most related words co-occur only with a small fraction of the extracted patterns. This problem necessitates some form of a dimensionality reduction in practice. Besides, such pattern-based methods require two words to co-occur in a specified window to extract patterns and thus to represent the pair. However, not every related pair co-occurs even in a large corpus (as shown in Figure 1.1). Therefore, pattern-based approaches fail to handle such unobserved but related words. Co-occurrences of word-pairs in sentences might also not be relevant for characterising the considered relation, an issue that typically leads to noisy relation representations.

Another limitation of this approach is **scalability**. For binary relations (relations between two arguments), the representation size in pair-pattern matrix grows quadratically with the number of words in the vocabulary. Therefore, it is computationally costly, especially if the vocabulary size is very large ($> 10^6$) and new words are continuously proposed because for each new word we must pair it with existing words in the vocabulary. Furthermore, a continuously increasing set of patterns is required to cover the relations that exist between the two words in each of those word-pairs.

In addition to the drawbacks mentioned above, the pattern-based approach loses **generalisation** ability. The majority of pattern-based methods assign each word-pair a representation and would not be able to generate relation representations for newly added word-pairs.

To overcome the limitations of the pattern-based approach, an alternative methodology that does not rely on pair-pattern co-occurrences is required. Such alternative methods must be able to represent the semantic relations that exist between all possible pairings of words requiring only semantic representations for the constituent words. These sort of methods are referred to as *compositional* in this thesis. A reason for this naming is because the compositional approach manipulates pre-trained word embeddings learnt from co-occurrence statistics in a corpus by applying operators that compose word representations to relation representations. The compositional approach has been inspired by the success of a new-family of word embeddings in solving analogies (as introduced in Section 1.2). The next section introduces word embedding models and relational reasoning in word embedding spaces.

2.5 Relational Reasoning with Word Embeddings

As mentioned earlier in Chapter 1, the work in this thesis seeks to avoid the problems in the pattern-based approach for representing relations between words, particularly sparseness, by adopting compositional methods for representing relations. This section will introduce our motivation for considering such an alternative for representing relations. The section commences by laying out dominant word representation models in Section 2.5.1 before moving to examine how relational information can be captured from the learnt word representations in Section 2.5.2. Then, Section 2.5.3 reviews some theoretical insights that explain the behaviour of word embeddings in solving analogies. Limitations of relational reasoning with word embeddings are discussed in Section 2.5.4. Finally, Section 2.5.5 presents efforts made to employ word embeddings for various NLP tasks in which relational reasoning is required.

2.5.1 Word Embedding Models

Representing the meaning of individual words in a semantic space has been studied extensively since 1996 (Landauer and Dumais, 1997; Lund and Burgess, 1996). Capturing the meaning of words plays a vital role in the most important advances of NLP applications. In the 1950s, a linguist called John Firth stated that words borrow their meanings from other words. Succinctly, this is called the distributional hypothesis of meaning, which states that "*you shall know a word by the company it keeps*" (Harris, 1954; Firth, 1957). This theory of meanings is extremely profound and widely applied in the NLP community. Let us consider the following example to clarify the idea of this hypothesis.

Question: X is a device that is easy to carry around; you can speak using X and use the Internet. What could X be?

a) a dog

b) an aeroplane

c) an iPhone

d) a banana

Here, even without knowing the meanings of the words in the candidate list, we can represent X by examining its contexts set (i.e., device, carry, speak, Internet, etc.), which functions as the clues that define the meaning of X. Thus, the correct answer for X that fits the given contexts is an iPhone. The distributional hypothesis implies that linguistic items (such as words in our context here) that appear in similar distributions have similar meanings.

In practice, existing word representation methods apply the distributional hypothesis in various ways on a large text corpus to teach computerised systems the meaning of words. Typically, each word is represented in term of its surrounding lexical contexts, and semantically similar words have comparable representations. Having representations of individual words is important for an unlimited number of computational linguistic tasks, including word sense disambiguation (McCarthy, 2007; Alsuhaibani and Bollegala, 2018), relation representation and classification (Glavaš and Vulić, 2018; Anke et al., 2019; Hakami and Bollegala, 2019a) and compositional semantics (Socher et al., 2013b), to name a few. If a model captures the meanings of individual words properly, we can then use compositional approaches to construct the meanings of other constituents such as phrases, sentences, documents and even relations as in the case of this thesis.

Word representations are generated by diverse methods. *Counting-based* is a classical approach for representing words in a vector space, and the generated vectors are referred to as distributional word vectors. Distributional representations are generally based on counting co-occurrences between words in a large text corpus. Specifically, a word vector represents the meaning of a word by a potentially high-dimensional sparse vector, where each dimension corresponds to a particular word that co-occurs with the word under consideration in some context. The counting process creates a word-context co-occurrence matrix such that words that co-occur with similar contexts will obtain a similar distribution in the matrix. The counting-based approach suffers from various issues, such as the imbalance of word frequencies due to rare and frequent words, which is addressed by weighting the raw values using an association measure such as Point-wise Mutual Information (PMI) or a log-likelihood ratio (Church and Hanks, 1990). Another issue associated with this approach is the high-dimensional space due to large vocabulary size. To combat the curse of dimensionality in the co-occurrence matrix ($\approx \mathbf{X} \in \mathbb{R}^{10^5 \times 10^5}$), dimensionality reduction techniques are applied to the matrix to generate denser representations while preserving the similarity between the two representations. The most widely used and effective dimensionality reduction methods are SVD, Principle Component Analysis (PCA) and Nonnegative Matrix Factorisation (NMF). This process of obtaining semantic representations of words by applying such reduction techniques on the co-occurrence matrix is termed Latent Semantic Analysis (LSA) (Deerwester et al., 1990).

Recently, a new family of word representation (a.k.a. embedding[1]) methods that are *prediction-based* have gained attention due to advances in neural networking techniques. Rather than counting co-occurrences between words, each word w is assigned a low-dimensional vector ($10 \sim 1000$ parameters) of random numbers; its parameters are then learnt such that we can accurately predict the words that appear in the same context as w. Compared to the observed contextual features of distributional representations, prediction-based methods yield to distributed representations with latent features that jointly represent the space efficiently. Typical models of distributed representations come from natural language modelling (Bengio et al., 2003; Collobert et al., 2011; Mikolov et al., 2013c).

The most popular and successful word embedding models of the prediction-based approach are Continuous Bag-of-Words (CBOW) and Skip-Gram (SG) (Mikolov et al., 2013b,a), Global Vector Prediction (GloVe) (Pennington et al., 2014) and FastText (Bojanowski et al., 2017). Similar to counting-based methods, these models require a large text corpus to learn efficient representations in an unsupervised fashion because the corpus is basically unannotated. In other words, there is no need for hand-labelled supervision to train prediction-based word embedding models. The following sections describe in detail the prediction-based word embedding models utilised in this thesis.

CBOW and SG models

CBOW and SG models learn word representations by considering the task of predicting words that co-occur in a local contextual window. Whereas CBOW is learnt to predict a target word given its context, the SG model predicts the surrounding window of context words of the target word. The two models are widely referred to as Word2Vec, which is a tool used for learning CBOW and SG word vectors.

The CBOW model is described briefly as follows. Initially, each word w in the vocabulary is assigned with two vectors one as a target (denoted as $\boldsymbol{w}$) and the other as a context word (denoted as $\tilde{\boldsymbol{w}}$). Given a sentence "*I had bread and <u>butter</u> for breakfast*", suppose we are interested in learning a semantic representation for the word *butter* (i^{th} target word w_i) considering the set of context words $C = \{w_{i-1}, w_{i-2}, w_{i+1}, w_{i+2}\}$ in a window of size $l = 2$ to left and right sides. The aim is to learn vectors encoding that given the context words {*I, had, bread, breakfast*}, it is more natural that the missing target word is *butter* rather than the word *pen*, for instance. To do so, the model seeks to maximise the probability of predicting *butter* among other words in the vocabulary. The likelihood of the word *butter* co-occurring in the given context words ($p(w_i \mid C)$) can be modelled using the dot product between corresponding word vectors normalised to be in a range $[0, 1]$. The word embeddings

[1] The terms *representation* and *embedding* are used interchangeably throughout this thesis to refer meaning representations.

of the context words in a specified window are averaged in the CBOW model, ignoring word order. However, it is computationally expensive to consider all the words in the vocabulary when predicting a target word. Thus, to reduce the complexity of such computations, two different training algorithms have been proposed, namely, hierarchical softmax and negative sampling. The CBOW objective function to be maximised with negative sampling is defined in (2.1).

$$J = \frac{1}{T}\sum_{i=1}^{T}\left(\log\sigma\left(\boldsymbol{w}_i{}^{\top}\sum_{-l\leq j\leq l}\tilde{\boldsymbol{w}}_{i+j}\right) + \sum_{w'\in S(\mathcal{V})}\log\sigma\left(-\boldsymbol{w'}^{\top}\sum_{-l\leq j\leq l}\tilde{\boldsymbol{w}}_{i+j}\right)\right) \tag{2.1}$$

Here, T is the number of tokens in the corpus, $\mathcal{V}$ is the vocabulary of words and $S(\mathcal{V})$ is the set of negatively sampled target words from $\mathcal{V}$. While parsing the text, the defined objective would maximise the probability of predicting the observed target word in the given context, whereas minimising the probabilities of negative words in that context. The SG is a reverse of the CBOW model as it considers $p(w_{i+j} \mid w_i)$, which is the probability of observing the context word w_{i+j} given the i^{th} target word w_i. This probability will be computed for each context word around w_i in a window of size l (i.e., $-l \leq j \leq l, j \neq 0$).

GloVe model

The global vector prediction model combines the properties of counting- and predicting-based methods. As indicated by the model name, GloVe takes advantage of global co-occurrence statistics between words while predicting their embeddings instead of local co-occurrences as in CBOW and SG. Specifically, GloVe first builds a co-occurrence matrix between words and then learns embeddings for the words such that by using the inner product between the corresponding embeddings, we can approximate the logarithm of the co-occurrence counts between the words. The least-square objective function of the GloVe to be minimised is defined in (2.2).

$$J = \sum_{i,j=1}^{|\mathcal{V}|} f\left(X_{ij}\right)\left(\boldsymbol{w}_i{}^{\top}\tilde{\boldsymbol{w}}_j + \boldsymbol{b}_i + \tilde{\boldsymbol{b}}_j - \log X_{ij}\right)^2 \tag{2.2}$$

Here, $\boldsymbol{w_i}$ is the vector representing the i^{th} target word, $\tilde{\boldsymbol{w_j}}$ is the vector for the j^{th} context word, and $|\mathcal{V}|$ is the number of words in the vocabulary. $\boldsymbol{b_i}$ and $\tilde{\boldsymbol{b_j}}$ are biases vectors associated with each target and context words. The co-occurrence matrix is denoted by $\mathbf{X}$, where X_{ij} is the number of times w_i co-occurs with w_j in the corpus (only nonzero elements are considered while training). f is a weighting function that aims to reduce the impact of frequent co-occurrences in $\mathbf{X}$. The weighting function f is parametrised by x_{max} (normalisation factor) and α (for nonlinearity), and defined in a way to be in a range $[0, 1]$

as follows:

$$f(x) = \begin{cases} (x/x_{max})^{\alpha} & \text{if } x < x_{max} \\ 1 & \text{otherwise} \end{cases} \tag{2.3}$$

Earlier described word embedding models (CBOW, SG and GloVe) are considered context-free because, by the end of the training, each word is assigned a single representation considering its distribution among the entire corpus. Recently, from 2018 onwards, the interest in this field has been directed towards what is called *contextualised* word representation models. After pre-training such models on a corpus considering different language modelling tasks, a representation is given to a word based on a given contextual window (e.g., sentence) in which the target word appears. For instance, the word *bank* would have two embeddings for each of the two given sentences: “I accessed the *bank* account” and “I waded to the *bank* and picked up my shoes”. In light of the previous example, contextualised embeddings can handle the polysemy issue. ELMo Embeddings from recurrent neural network Language Models (Peters et al., 2018) and BERT Bidirectional Encoder Representations from Transformers (Devlin et al., 2019) are pioneering models of contextualised word embeddings that are fine-tuned successfully on variety of NLP downstream tasks such as question answering and lexical entailment. It is worth noting that at the time of working for this thesis, contextualised word embedding models had not emerged yet. Moreover, the study of this thesis seeks to model a target relation between words without requiring the relational patterns at inference time to avoid the sparsity problem; thus contextualised word embeddings were not of our interest at this stage.

Having understood word embeddings and how they are generated, the next section moves on to discuss relational reasoning with pre-trained word embeddings.

2.5.2 Relations as PairDiffs: A Compositional Approach

Generally speaking, the earlier-mentioned word embedding models seek to map semantically similar words to nearby points in semantic space of representations. A proof of this concept is that word representations are primarily evaluated using a semantic similarity task, which measures the attributional similarities between words and compares with gold-standard human scores. Several word similarity datasets, such as WS353 (Finkelstein et al., 2002), MC (Miller and Charles, 1991) and MEN (Bruni et al., 2012), have been established as benchmarks for predicting the semantic similarity between words. Two words that share an identical set of attributes (such as *car* and *automobile*) should have a high degree of attributional similarity and thus can be considered to be synonyms. However, two concepts can also be similar according to a different notion of similarity. As noted by Hill et al. (2015), these benchmarks include word-pairs with high scores because they are associated or related by a specific relation type. For instance, in the WS353 dataset, (*media*, *radio*) and (*cup*, *coffee*) are assigned values of 7.42 and 6.58, respectively. The two words in these pairs are not purely attributionaly similar, but they are related by either Hypernym as in (*media*, *radio*) or content-container as in (*cup*, *coffee*).

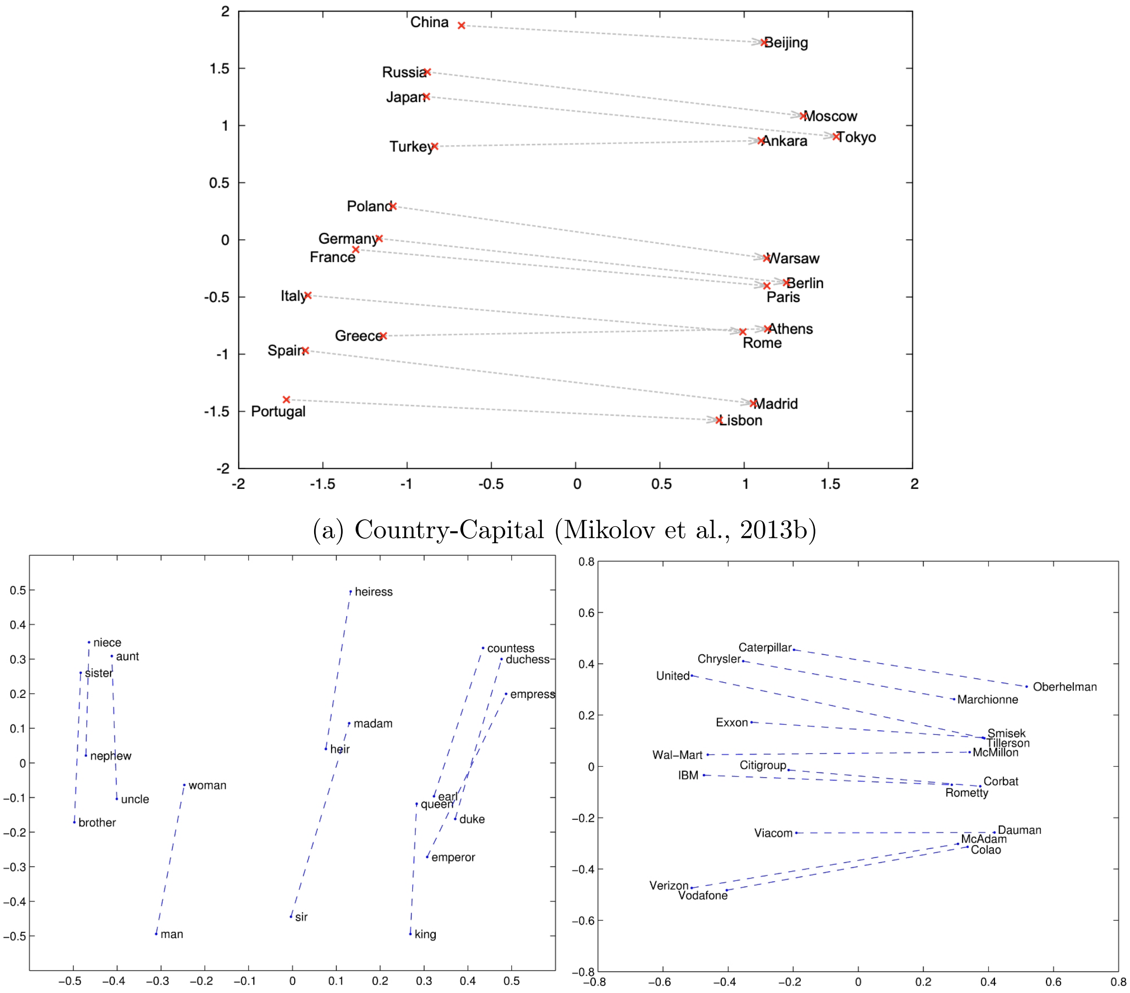
Research in word representations moved to evaluate such representations using a different notion of similarity between words, referred to as a relational similarity (Turney, 2006; Jurgens et al., 2012). For example, rather than focusing on how similar *France* is to *Italy* as countries, they wanted to assess whether *Paris* is related to *France* in the same way that *Rome* is related to *Italy*. They convert these four words into the following question: "What is the word that is related to *Italy* in the same way as *Paris* is related to *France*?". This analogy question between two word-pairs is formally written as $France : Paris :: Italy :?$, where the correct word to complete such an analogy is *Rome* because *Paris* is the capital-of of *France* as *Rome* is the capital-of *Italy*. In addition to the semantic kind of analogies as in the previous example, analogies of morphological forms of words show another interesting way of associations between words. For example, *kings* to *king* as *queens* to *queen*, *small* to *smallest* as *big* to *biggest*. For this sort of evaluation, several benchmarks have been proposed, such as MSR syntactic analogies (Mikolov et al., 2013c) and Google analogies (Mikolov et al., 2013a).

The hypothesis that will be tested while answering analogical questions is whether semantic spaces of pre-trained word embeddings encode implicit relationships between words. Mikolov et al. (2013c), in their widely cited work, show remarkable success in demonstrating a relationship between two words by subtracting the two corresponding word vectors. In other words, the vector offsets of word-pairs that exemplify a particular relationship are almost parallel. Mikolov et al. (2013c) refer to this property as linguistic regularities in the semantic spaces and they illustrate such regularity as presented in Figure 1.2 that shows Opposite-Gender and Singular-Plural word-pairs, which can be mathematically written as:

$$\boldsymbol{woman} - \boldsymbol{man} \approx \boldsymbol{queen} - \boldsymbol{king} \approx \boldsymbol{aunt} - \boldsymbol{uncle}$$
$$\boldsymbol{kings} - \boldsymbol{king} \approx \boldsymbol{queens} - \boldsymbol{queen}$$

We denote this operation as PairDiff throughout this thesis, which means the difference between two vectors in a pair. Hence, to answer the question $man : woman :: king : d$, PairDiff is applied for each pair in the two sides of the analogy to find the missed word d as in (2.4). Then, using the nearest neighbour search between $\boldsymbol{d}$ and embeddings of other words in the vocabulary considering cosine similarity scores, the closest word to d is selected as the answer.

$$\begin{aligned} \boldsymbol{woman} - \boldsymbol{man} &= \boldsymbol{d} - \boldsymbol{king} \\ \boldsymbol{woman} - \boldsymbol{man} + \boldsymbol{king} &= \boldsymbol{d} \end{aligned} \tag{2.4}$$



(a) Country-Capital (Mikolov et al., 2013b)

(b) Male-Female (Pennington et al., 2014)

(c) Company-CEO (Pennington et al., 2014)

Figure 2.4: Tow-dimensional visualisation for selected word-pairs of pre-trained SG (a) and GloVe embeddings (b and c). These figures are taken from the original papers.

As presented in Figure 2.4, visualisations of two-dimensional projection for selected word-pairs of SG and GloVe embeddings emphasise the regularity of word-pairs that belong to the same relation under the PairDiff method. Word representation models that are proposed after the emergence of this property adopt solving word analogical questions as a new intrinsic evaluation method (Schnabel et al., 2015). This remarkable property of word embeddings also sparked a renewed interest in methods that compose relational embeddings from word embeddings, as in the case of this thesis. This approach for representing relations between words is called *compositional*, throughout the thesis, because the way in which the relation representation is composed using the semantic representations of the constituent words avoiding the need for relational patterns.

Intuitive explanation of PairDiff. Let us think about word embeddings and then consider their difference to represent relations. For low-dimensional word embeddings such as CBOW, SG and GloVe, dimensions are latent in the sense that they do not correspond to known features. A collection of d dimensions altogether defines the meaning of a concept from co-occurrence information, as described in Section 2.5.1. In this regard, the embedding vector $\boldsymbol{king}$ represents the word *king* in terms of its most salient features within the dimensions of the vector. Thus, $\boldsymbol{king} \in \mathbb{R}^d$ (d usually set to be in a range $[50, 1000]$) has high values in dimensions corresponding to *royalty*, *masculinity* and any feature related to *humanity*. On the other hand, *royalty* dimensions are assigned low values in the vector of *man*, whereas we expect high values for *masculinity* and *humanity* features as in $\boldsymbol{king}$. Intuitively, subtracting $\boldsymbol{man}$ from $\boldsymbol{king}$ would cancel out[2] the features of *man* that are *king*-specific, and retaining the features that distinguish a royalty. Similarly, we can think about the operation of $\boldsymbol{queen} - \boldsymbol{woman}$. This is how PairDiff derives relational representations from word embeddings.

However, NLP researchers considered the ability to reason about relations by manipulating pre-trained word embeddings as being purely coincidental because:

- word embeddings are not explicitly trained to capture such relational information between words; and
- when Mikolov discovered this remarkable property, there was no theoretical understanding of why this has to happen.

In response to this, there have been some papers that attempt to provide formal analysis to understand this interesting property in word semantic spaces, as will be seen in the next section.

2.5.3 Theories of Word Embedding Geometrics for Word Analogies

The first attempt that explains regularities for word analogies was with the release of GloVe word embeddings. Due to the fact that the GloVe model works on factorising global co-occurrence statistics between words, their intuition was that some aspect of meaning for two words could be inferred from the ratio of co-occurrence probabilities. To explain this, we can take the dominant word-pair (*man*, *woman*) as an example. The relation between *man* and *woman* can be expressed as the probability ratio $\frac{p(w|man)}{p(w|woman)}$ among various words w in the vocabulary. For gender-related words w, the ratio will be large for masculinity words (*he*) and small for femininity words (*she*), whereas for gender-neutral words the ratio would be almost close to 1. The ratio scores allow discriminating relevant from irrelevant

[2] It is unlikely a dimension has the same value in both words, so the difference vector will still have some nonzero value left in the dimension.

contexts from the two target words, and thus the ratios of related word-pairs are expected to be similar as in (2.5).

$$\frac{p(w \mid man)}{p(w \mid woman)} \approx \frac{p(w \mid king)}{p(w \mid queen)} \approx \frac{p(w \mid uncle)}{p(w \mid aunt)} \tag{2.5}$$

Pennington et al. (2014) designed the GloVe objective in (2.2) in a way that the ratio of co-occurrences between words is encoded in the word vector space using vector differences. Although explicitly designing the word space in such a way improved the performance of the word analogy task, the authors do not substantiate their conjecture with mathematical analysis or empirical validation.

There have been few notable formal explanations of why $\boldsymbol{king} - \boldsymbol{man} + \boldsymbol{woman} \approx \boldsymbol{queen}$ within word embedding spaces, starting with the latent variable text generative model (Arora et al., 2016), which is used to provide a theoretical analysis of the close approximation between the PMI co-occurrence matrix and its low-ranked SVD. The central assumption used in the theory of Arora et al. is the isotropy of word embeddings in a space, verified empirically in their paper. A critical implication from their proposed model and theory, in the extent of the compositional-based approach, is that relations represent lines in low-dimensional word embedding space. Namely, from the ratio of probability suggested by Pennington et al. (2014), they show that a relation is represented by a vector such that the PairDiff vectors for all word-pairs of this relation are similar to the relation vector with a small amount of noise. Ethayarajh et al. (2019) thoroughly analysed why and when word analogies can be solved using SG and GloVe word embeddings. They assign the success of linear analogies within a set of word-pairs to co-occurrence statistics between words over a training corpus used to learn word embeddings. Along with the same word-pair set example {(*king*, *queen*),(*man*, *woman*)}, they theoretically and empirically validate that a specialised PMI (termed co-occurrence shifted PMI) has to be the same for (*king*, *queen*) and (*man*, *woman*) in order for the differences $\boldsymbol{king} - \boldsymbol{queen}$ and $\boldsymbol{man} - \boldsymbol{woman}$ to be identical.

Differently, Allen and Hospedales (2019) explain why word embeddings should show analogies using a paraphrasing model under word transformations. They provide evidence for the linear combinations (additions and subtractions) of embeddings using two sets of words that are paraphrasing each other, which mean they have the same distributions over context words. For example, *king* paraphrase {*man*,*royal*}. Allen and Hospedales (2019) give a probabilistic definition of paraphrasing that is applied to justify why analogies hold. The analogy "*man* is to *king* as *woman* to *queen*" indicates the existence of a paraphrase between the two sets {*woman*, *king*} and {*man*, *queen*} through word transformations from *man* to *king* and from *woman* to *queen*. A transformation can be chosen to be parametrised by adding *king* and subtracting *man* to move from one word to another. Along a similar line, Gittens et al. (2017) explain word analogies by studying the additive compositionality in SG embedding space with a strong assumption that word frequencies are uniform in a corpus.

All the aforementioned theoretical understanding of why word analogies can be solved using pre-trained word embeddings is essential to eliminate surrounding queries for compositional-based methods for relations, which in turn support the conducted research in this thesis. However, these research studies focus on providing explanations that support linear combinations of word embeddings towards solving analogies, considering the success of the initial hype. The work conducted in this thesis goes beyond the scope of the linear additive combinations for relation representations in different ways. For instance, a theoretical analysis for a bilinear operator between word embeddings to represent relations for word-pairs is conducted, as will be presented in Chapter 4. Also, different models that learn parametrised compositional operators for relations between words using neural networks with nonlinearities are proposed in Chapter 5 of this thesis.

2.5.4 Criticisms of PairDiff Method

The PairDiff approach for word analogies has attracted a significant amount of attention in terms of validating its success for reasoning about relations empirically considering multiple views. Most prior work is devoted to analysing the ability of PairDiff towards completing word analogies as coined by Mikolov et al. (2013c), whereas some others test the utility of PairDiff vectors for learning relations (Levy et al., 2015b; Vylomova et al., 2016; Chen et al., 2017).

Investigating the power of PairDiff for answering analogy questions (i.e., $a : b :: c :$?), Linzen (2016) and Rogers et al. (2017) attribute a large percentage of the success of PairDiff to the lexical similarities between the words in the question. Returning to the well-known example in (2.4), they show that the correct answer to *man* : *woman* :: *king* :? is predicted to be *queen* because *queen* is the nearest neighbour to *king* in an embedding space regardless of the offset vector $\boldsymbol{woman} - \boldsymbol{man}$. Although PairDiff performs well on the Google analogy dataset, which is used initially for the evaluation of PairDiff, its performance for other relation types such as paradigmatic relations (Hypernym, Synonym, Antonym) has been poor (Köper et al., 2015; Gladkova et al., 2016). In response to these limitations, alternative methods have been proposed to resolve analogies, indicating that relational properties that cannot be extracted by one method from word embeddings can be accessed by another method (Levy and Goldberg, 2014; Drozd et al., 2016).

Even though answering analogies from word embedding motivates the work conducted in this thesis, it is worth noting that the focus is tended on representing features of relations that hold between two words within a given pair rather than analogy completion. In this regard, Vylomova et al. (2016) test the generalisation of the PairDiff vectors across different kinds of relations by evaluating the space of PairDiff in their own right under unsupervised (clustering) and supervised (classifying word-pairs to relation types) settings. Considering a broad coverage of relation types outside those in the Google dataset, they conclude that important information about relations is implicitly embedded in PairDiff vectors. However, they show that syntactic relations are clustered better than semantic relation types. Whereas Vylomova et al. (2016) show that word-pairs can be classified accurately to relations by training a supervised classifier on PairDiff vectors, Levy et al. (2015b) argue that the learnt method considers individual word properties such as a prototypical hypernym rather than learning relations between words. Similarly, Fu et al. (2014) report that the hypernym-hyponym link between words is more complicated, and a single offset vector cannot completely represent it.

The space of unsupervised operators that are proposed so far in the literature is limited in the sense that the operators are pre-defined and fixed, and they cannot be adjusted to capture the actual relations that exist between words. It is unrealistic to assume that the same operator can represent all relation types from the word embeddings learnt from different word embedding learning algorithms. On the other hand, there are many datasets such as SemEval 2012 Task2, Google analogies and MSR analogies that already provide examples of the relation types that exist between words. Motivated by this, it has been proposed to learn parametrised relational compositional operators using word embeddings, as will be presented in Chapter 5.

2.5.5 Learning Relational Tasks via Word Embeddings

Since the success of solving word analogy, as first proposed by Mikolov et al. (2013c), several efforts have been made to use unsupervised word embeddings for tasks in which accurately capturing relational features is critical to improving performance. These relational tasks include word analogies (Levy and Goldberg, 2014; Drozd et al., 2016; Bouraoui et al., 2018), relation classification (Attia et al., 2016; Glavaš and Ponzetto, 2017; Glavaš and Vulić, 2018; Wang et al., 2019b), hypernym generation (Fu et al., 2015; Yamane et al., 2016; Wang et al., 2019a) and bilexical prediction (Madhyastha et al., 2014, 2015; Gupta et al., 2017).

Most of the work in the literature tends to classify word-pairs into relation types from a pre-defined relation set via embeddings of words in the pairs. In a direct way, supervised classifiers have been learnt using distributional features of word-pairs to predict relation labels, wherein word-pairs (a, b) are presented to a classifier as a composition of a and b embeddings (Vu and Shwartz, 2018). Wang et al. (2019b) assign probability distribution over relation types for unlabelled word-pairs (a, b) by training a classifier on features taken from learnt relation-specific projection matrices from source word embeddings (i.e., a) to target word embeddings (i.e., b) of labelled training pairs. Then, they learn relation vectors (spherical relation embeddings as they express) such that these vectors can predict the neighbour pairs in a sequence of generated pairs. The proposed spherical relation vectors outperform multiple pattern-based methods for relation classification. Along similar lines, Glavaš and Ponzetto (2017) propose a dual tensor model to detect whether an asymmetric relation holds between word-pairs. Specifically, the dual tensor model learns two different specialisation tensors (for source and target words), the specialised word vectors are then mapped through a bilinear relational tensor to predict a confidence score for word-pairs.

Another task in which pre-trained word embeddings are applied successfully is generating a hypernym word for a given hyponym. Fu et al. (2015) construct a semantic hierarchy of concepts linked by is-a relation by linearly projecting hyponyms to their hypernym embeddings within a cluster of PairDiff vectors. Rather than clustering pairs using PairDiff vectors, Yamane et al. (2016) propose a model that jointly learns to cluster and project word-pairs.

For word-level bilexical prediction, Madhyastha et al. (2014) learn a low-ranked bilexical operator between the embeddings of words in pairs sharing a given relation so that they can predict a modifier for an unseen noun among a vocabulary, such as predicting *electronic* as a better modifier for *device* than *case*. Similarly, Gupta et al. (2017) study the ability of word embeddings for predicting embeddings for missing entities in a KG by training a nonlinear feed-forward neural network. Ethayarajh (2019) also learn linear and orthogonal transformations in word embedding spaces such that a word a can be transformed to b where a relation r holds between a and b.

This thesis addresses the more general task of learning compositional operators on word embeddings to obtain representations for relations between words. Unlike this approach, all the efforts mentioned above use word embeddings for specific tasks instead of broadly mapping word-pairs to relation embeddings. Having stand-alone representations for relational features between words can enrich features of word embeddings and thus improve the performance of a variety of downstream NLP tasks such as textual entailments, machine translation and question answering. Also, relation representations allow us to conduct all of the earlier mentioned tasks as the evaluations carried out for this thesis.

2.6 Complementarity of Pattern-based and Compositional Approaches for Relations

As elaborated in Section 2.4.4 and 2.5.4, both pattern-based and compositional approaches for representing relations suffer from significant drawbacks. While the pattern-based methods such as LRA efficiently leverage patterns that link related concepts, they fail with respect to representing unobserved related pairs. On the other hand, the compositional PairDiff method may able to represent such related but unseen pairs; still, they indirectly utilise the relational properties because they consider global contextual statistics for each word to learn relations. However, the two approaches have complementary properties when it comes to representing relations. Hence, there is a need for hybrid approaches that provide a balance between the data sparsity in the pattern-based methods and the lack of relational information in the compositional approach.

A limited number of studies have attempted to address this requirement for various relational tasks. To measure relational similarity, Zhila et al. (2013) combine heterogeneous models, including distributional word embeddings and lexical patterns. They show that the compositional method, which uses the PairDiff, reported encouraging results for many relation types in the SemEval-2012 task 2 dataset. Shwartz et al. (2016) and Shwartz and Dagan (2016) integrate dependency paths of relational patterns, which are encoded using recurrent neural networks, and distributional word embeddings to classify word-pairs to semantic relations. Integrated models boost the performance for the identification of semantic relations, in contrast to models that employ each source separately.

Few recent studies have been devoted to incorporate the two types of information to improve the relation representations (Washio and Kato, 2018b,a; Joshi et al., 2019). Washio and Kato (2018b) extend LRA through proposing an unsupervised relational operator that is learnt to make the compositional and pattern representations similar using a negative sampling training objective. During inference time, the proposed method does not need to access relational patterns for representing word-pairs. The authors also found that their proposal can be used worthily to predict missing dependency paths between word-pairs that do not co-occur in a corpus (Washio and Kato, 2018a). Joshi et al. (2019) show improving in performance when providing such word-pair embeddings, which are obtained from a hybrid compositional method trained on the two sources of information, to question answering and cross-sentence natural language inference models.

Previously described hybrid methods for relation representation specialise word vectors such that relational properties are encoded. Thus, they might fail to represent relations between unseen words. This thesis contributes towards hybrid approaches to represent relations by proposing a context-guided relation embedding method, as will be presented in Chapter 5 (Section 5.3). The parametrised operator we learn generalises in the sense that it can be applied to any new word-pair or relation type, and it is not limited to the words and relations that exist in the training data.

Parallel to exploiting texts as a source of relational knowledge, KGs represent another thread of work for relations. The section below reviews related literature to relational KGs and representations for KG entities and relations.

2.7 Multi-Relational Knowledge Representations

Let us now move from the expressive and unstructured text-based source of information for relations to another finger-structured source referred to as knowledge graphs. Chapter 7 in this thesis is directed at representing relations between entities in the context of structured KGs. The remainder of this section is organised as follows, it begins by reviewing the structure of KGs in Section 2.7.1. Then, knowledge graph embedding methods for inferring missing links within a graph are descirbed in Section 2.7.2.

2.7.1 Knowledge Graphs

KGs organise the knowledge that we have about entities and the relations that exist between entities in the form of labelled graphs, where entities are denoted by the vertices and the relations are denoted by the edges that connect the corresponding entities. A KG can be represented using a set of relational tuples of the form (h, r, t), where the relation $r \in \mathcal{R}$ exists between the (head) entity $h \in \mathcal{V}$ and the (tail) entity $t \in \mathcal{V}$ such that the direction of the relation is from h to t. Here, $\mathcal{V}$ and $\mathcal{R}$ respectively denote the sets of entities and relations in the KG. For example, the relational tuple (*Donald Trump*, president-of, *US*) indicates that the president-of relation holds between *Donald Trump* and *US*. Freebase (Bollacker et al., 2008) and WordNet (Miller, 1995) are widely used KGs. Freebase is an example of social KG of relations between named entities (people, places, etc.), whereas WordNet is a semantic graph of linguistic relations between words. Other KGs are domain-specific such as AceKG for academic publications (Wang et al., 2018), Product KG at Amazon for e-commerce and the UMLS semantic network for the biomedical field. Organising data in graphs benefits a variety of information extraction and NLP tasks such as relational search (e.g., Google KG), natural language generation (Koncel-Kedziorski et al., 2019; Logan et al., 2019) and question answering (Das et al., 2017; Sydorova et al., 2019).

Traditionally, KGs were created manually by specialists in the domain. However, the cost of constructing such large-scaled graphs is high in terms of time, money and effort for specialised fields such as biomedicine. Recently, with the advent of machine learning techniques, automated methods have been proposed to extract relational triples from unstructured texts and thus help enlarge KGs. Many proposed methods are devoted to naming entities in unstructured texts and then predicting relations between them automatically (Mintz et al., 2009; Getoor and Machanavajjhala, 2012; Konstantinova, 2014; Riedel et al., 2013; Ren et al., 2017; Bosselut et al., 2019). The embedding-based approach is another technique to increase KGs coverages by embedding KGs into numerical objects such that reasoning and inference are applied to predict missing links and classify triples. More details about knowledge graph embedding methods are discussed in the following section.

Table 2.1: Score functions proposed in selected prior work on KGEs. Entity embeddings $\boldsymbol{h}, \boldsymbol{t} \in \mathbb{R}^d$ are vectors in all models, except in ComplEx where $\boldsymbol{h}, \boldsymbol{t} \in \mathbb{C}^d$. Here, $\ell_{1/2}$ denotes either ℓ_1 or ℓ_2 norm of a vector. In ComplEx, $\bar{\boldsymbol{t}}$ is the element-wise complex conjugate.

KGE method	Score function $f(h,r,t)$	Relation parameters
Unstructured(Bordes et al., 2011)	$\|\|\boldsymbol{h} - \boldsymbol{t}\|\|_{\ell_{1/2}}$	none
Structured Embeddings SE (Bordes et al., 2011)	$\|\|\mathbf{R}_1\boldsymbol{h} - \mathbf{R}_2\boldsymbol{t}\|\|_{\ell_{1,2}}$	$\mathbf{R}_1, \mathbf{R}_2 \in \mathbb{R}^{d \times d}$
Translating Embeddings TransE (Bordes et al., 2013)	$\|\|\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}\|\|_{\ell_{1/2}}$	$\boldsymbol{r} \in \mathbb{R}^d$
DistMult (Yang et al., 2015)	$\langle \boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t} \rangle$	$\boldsymbol{r} \in \mathbb{R}^d$
RESCAL (Nickel et al., 2011)	$\boldsymbol{h}^\top \mathbf{R} \boldsymbol{t}$	$\mathbf{R}^{d \times d}$
ComplEx (Trouillon et al., 2016)	$\langle \boldsymbol{h}, \boldsymbol{r}, \bar{\boldsymbol{t}} \rangle$	$\boldsymbol{r} \in \mathbb{C}^d$

2.7.2 Knowledge Graph Embeddings

Despite the best efforts to create complete and large-scale KGs, most KGs remain incomplete and do not represent all the relations that exist between entities (Min et al., 2013). In particular, new entities are constantly being generated, and new relations are formed between new as well as existing entities. Therefore, it is unrealistic to assume that a real-world KG would be complete at any given time point. Developing approaches for KG completion is an important research field associated with KGs.

Analogous to the word embedding methods discussed in Section 2.5.1, KG components can be embedded into numerical formats. KGE methods learn representations (also referred to as embeddings as in the case of word embeddings) for the entities and relations in a given KG (Bordes et al., 2011; Nickel et al., 2011; Yang et al., 2015; Nickel et al., 2016; Trouillon et al., 2016). The learnt KGEs can be used for *link prediction*, which is the task of predicting whether a particular relation exists between two given entities in the KG. Specifically, given KGEs for entities and relations, in link prediction, we predict r that is most likely to exist between h and t according to some scoring formula. Thus, by embedding entities and relations that exist in a knowledge graph in some (possibly lower-dimensional and latent) space, we can infer previously unseen relations between entities, thereby expanding a given KG.

Knowledge Graph Embedding Process

At a high-level of abstraction, KGE methods can be seen as differing in their design choices for the following two main problems:

(a) how to represent entities and relations using some mathematical entities in linear algebra, and

(b) how to model the interaction between two entities and a relation that holds between them.

Next, we briefly discuss prior proposals to those two problems.

A popular choice for representing entities is to use vectors, whereas relations have been represented by vectors (e.g., Translating Embeddings and DistMult), matrices (e.g., Structured Embeddings and RESCAL) or by 3D tensors as in the Neural Tensor Network model proposed by Socher et al. (2013a). ComplEx (Trouillon et al., 2016) introduced complex vectors for KGEs to capture the asymmetry in semantic relations. Given entity and relation embeddings, a scoring function is defined that evaluates the strength of a relation r between two entities h and t in a triple (h, r, t). The scoring functions that encode various intuitions have been proposed such as the ℓ_2 norm of the vector formed by a translation of the head entity embedding by the relation embedding over the target embedding, or by first performing a projection from the entity embedding space to the relation embedding space (Yoon et al., 2016). As an alternative to using vector norms for scoring functions, DistMult and ComplEx use the component-wise multi-linear dot product. Lacroix et al. (2018) proposed the use of nuclear 3-norm regularisers instead of the popular Frobenius norm for canonical tensor decomposition. Table 2.1 shows the scoring functions along with algebraic structures for entities and relations proposed in selected prior work in KGE learning.

Once a scoring function is defined, KGEs are learnt that assign better scores to relational triples in existing KGs (positive triples) over triples where the relation does not hold (negative triples) by minimising a loss function such as the logistic loss (RESCAL, DistMult, ComplEx) or marginal loss (TransE). Because KGs record only positive triples, a popular method to generate pseudo negative triples is to perturb a positive instance by replacing its head or tail entity by an entity selected uniformly at random from the vocabulary of the entities $\mathcal{E}$. However, uniformly sampled negative triples are likely to be obvious examples that do not provide much information to the learning process and can be detected by simply checking for the type of the entities in a triple. Cai and Wang (2018) proposed an adversarial learning approach where a *generator* assigns a probability to each relation triple and negative instances are sampled according to this probability distribution to train a *discriminator* that discriminates between positive and negative instances.

In link prediction and triple classification (predicting whether a triple is true or false) benchmark tasks, impressive results are reported for state-of-the-art KGE methods, with an 88.8% classification accuracy being achieved for the FB13 benchmark (Nguyen et al., 2018). Nevertheless, existing KGE models have some limitations as will be discussed in the following section.

2.7.3 Limitations of Existing KGE Models

Despite the good empirical performances of the existing KGE methods, the existing scoring functions are heuristically motivated to capture some geometric requirements of the embedding space. Theoretical understanding of KGE methods is comparatively underdeveloped. For example, it is not clear how the heuristically defined KGE objectives relate to the generative process of a KG. This thesis endeavours to fill this gap by providing a theoretical analysis of KGE. Specifically, a generative model of KGs that derives a relationship between $p(h, t \mid r)$ (the probability of r holding between h and t) and the embeddings of r, h and t is proposed. Then, we propose a learning objective to learn KGEs from a given KG such that the relationship given by our proven theorem is empirically satisfied. Our analysis is an extension of the random walk model proposed by Arora et al. (2016) on word embeddings to KGEs. The developed theoretical model is introduced in Section 6.2.

Noteworthy, the links that can be predicted using all previously mentioned KGEs are confined to $\mathcal{R}$, which is the set of relation types that already exists in the KG. In other words, we cannot predict novel relation types using the pre-trained KGEs alone. Relatively, there is a general paucity of studies that seek to predict representations for novel relations. Ma et al. (2019) propose TransW that extends the TransE model by predicting embeddings for unseen relations as well as entities from their word embeddings and thus detecting unknown facts. This thesis contributes towards solving this problem differently, as will be presented in Section 6.3. In short, considering the fact that relations that exist in a KG are often closely related (e.g., $\mathsf{born\_in} \wedge \mathsf{capital\_of} \rightarrow \mathsf{nationality}$), *relation composition* is proposed, which is a task of forming an embedding for a new relation type from given relation embeddings. Several studies have exploited such latent correlations between relations using multi-hop links between entities (i.e., $h \xrightarrow{r_1} e_1 \xrightarrow{r_2} t$) while learning representations for a KG, rather than only considering direct links ($h \xrightarrow{r} t$) (Lin et al., 2015; Nathani et al., 2019). For example, Lin et al. (2015) propose Path-based TransE (PTransE) that modify TransE objective to be $||\boldsymbol{h} + g(\boldsymbol{r_1}, \boldsymbol{r_2}) - \boldsymbol{t}||_{\ell_{1/2}}$, where g is a composition function that outputs a composed vector for the path. PTransE specifically tunes TransE by jointly learning entity, relation and path representations, and has been evaluated on existing relations as other typical KGE methods. However, our proposed relation composition is universal as they are not parametrised by the entities or relations in a KG.

2.8 Summary

This chapter reviewed relevant background material regarding the task of representing relations between words considering two different resources, namely text corpora and KGs. At a text corpus level, related work for representing relations was reviewed which broadly falls into two approaches: pattern-based, which exploits co-occurrence linguistic contexts of pairs, and compositional, which apply operators on word embeddings. How compositional methods can resolve the data sparsity drawback in the pattern-based approach was particularly discussed as it forms a motivation behind most of the work in this thesis. Studies that criticised the unsupervised PairDiff method for reasoning about relations were also presented. The last part of the chapter presented KG resources of structured relational knowledge. KGE methods were introduced as an efficient methodology to combat sparsity in KGs.

The chapter that follows moves on to consider the task of deriving relational features from word-level representations using unsupervised compositional operators.

3 Deriving Relational Features from Word Representations

3.1 Introduction

Chapter 2 outlined two approaches for learning relation representations for word-pairs, namely pattern-based and compositional approaches. Recall that the compositional methods access relational information between the words using the features in their word embeddings, which overcomes the sparsity problem in the pattern-based methods. Features that represent words are typically obtained by applying deep learning methods on a large corpus of text considering co-occurrences between words either in counting- or prediction-based fashion. It has been shown that the feature set that encodes word semantics includes features that are effective to induce relations between words. The first serious discussions of extending lexical semantics of words to capture relations between words emerged when Mikolov et al. (2013c) demonstrated linguistic regularities in the vector space of prediction-based word embeddings. Specifically, the authors found that the relationship between two words can be characterised by the vector offset of the corresponding word embeddings. The well-known example is the gender-direction relationship of the two word-pairs: (*man*, *woman*) and (*king*, *queen*), where the offset vectors of $\boldsymbol{woman} - \boldsymbol{man}$ and $\boldsymbol{queen} - \boldsymbol{king}$ are shown to be approximately parallel. This finding sparked a renewed interest in methods that derive relational features for word-pairs from word embeddings of the related words.

Before the advent of prediction-based word embeddings, Turney (2012) represented words in a way such that semantic relations between words can be modelled. Specifically, he constructs two spaces that represent words, namely domain and function spaces which respectively consisted of nouns and verbs. The author then modelled semantic relations between two word-pairs (a, b) and (c, d) by measuring domain similarities between a and b (likewise for c and d), and functional similarities between a and c (likewise between b and d). For instance, the two word-pairs (*carpenter*, *wood*) and (*mason*, *stone*) show that each pair has a relatively high domain similarity. At the same time, *carpenter* and *mason* have high functional similarity because both are *artisans*, while *wood* and *stone* share the function of *materials*. Thus we can infer that (*carpenter*, *wood*) is relationally similar to (*mason*, *stone*).

Motivated by the studies discussed above, this chapter aims to derive relation representations from the features that represent individual words in different ways. A systematic study for various compositional operators that can be applied under unsupervised settings on word embeddings to obtain relation features is conducted. Also, inspired by the heuristically constructed domain and function spaces in Turney (2012) work, we consider the observed contextual features from counting-based word representations to propose a data-driven approach for discovering representative features for relational similarity measurement.

The remainder of this chapter is organised as follows. We first present the word embedding models that are used in this chapter, and throughout the thesis as well, in Section 3.2. In Section 3.3, benchmark datasets that are used to evaluate relation representations are introduced. Then, Section 3.4 is devoted to the conducted study for unsupervised compositional methods for representing relations from word-level representations. The task of discovering representative feature spaces from counting-based word embeddings to measure the relational similarity is discussed in Section 3.5. The chapter is concluded with a summary in Section 3.6.

3.2 Training Word Embeddings

The study carried out in this thesis for obtaining relation representations assumes the availability of pre-trained word embeddings. As discussed in Section 2.5.1, deep learning models have been exploited to learn features that represent words using a large collection of text. A wide range of models have been proposed to obtain representations for words. In this chapter (and throughout the thesis), the three widely used prediction-based word embedding methods are considered, namely CBOW, SG and GloVe. Section 2.5.1 introduced these word embedding models in detail. For consistency of the comparison, all word embedding learning methods are trained on the same ukWaC corpus[1], which is a web-derived corpus in English consisting of ca. 2 billion words (Ferraresi et al., 2008). We lowercase all the text and tokenise using the NLTK tool[2], and we use the publicly available implementations by the original authors of CBOW, SG[3], and GloVe[4] for training the word embeddings with the recommended parameters settings. Specifically, in GloVe, the co-occurrence weighting parameters x_{max} and α are respectively set to 100 and 0.75, the maximum iteration is 50, and the contextual window size is equal to 15 words before and after the target word. In

[1] http://wacky.sslmit.unibo.it/doku.php?id=corpora

[2] http://www.nltk.org

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

[4] http://nlp.stanford.edu/projects/glove/

CBOW and SG, the context window is set to eight words, the negative sampling rate is set to 25 words for each co-occurrence, 15 iterations, and sampling parameter equal to 10^{-4}. The vocabulary is restricted to the words that appeared more than six times in the corpus, resulting in 1,371,950 unique words. Using each of the word embedding learning methods, we train 300-dimensional word embeddings.

In addition to prediction-based word embeddings described above, counting-based word representations are also considered in our study. This method assigns each word with a high-dimensional vector that captures the contexts in which it occurs. Unigram counts from the ukWaC corpus are first constructed. The co-occurrences between low-frequency words are rare and result in a sparse co-occurrence matrix. To avoid this issue, the most frequent 50,000 words in the corpus are used as our vocabulary and co-occurrences between only these words considered. We found that a vocabulary of 50,000 frequent words is sufficient for covering all the benchmark datasets used in the evaluations. Moreover, truncating the co-occurrence matrix to the top frequent contexts makes the dimensionality reduction methods computationally inexpensive. Then, the word-context co-occurrence statistics are computed from the corpus using windows of size five tokens on each side of the target word. We weight the co-occurrences by the inverse of the distance between the two words measured by the number of tokens appearing between the two words. Afterwards, the Positive PMI (PPMI) is computed from the co-occurrence matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$ (n words in the vocabulary and m contexts in which they are the same in our settings) as follows:

$$\mathrm{PPMI}(x, y) = \max\left(0, \log \frac{p(x, y)}{p(x)p(y)}\right), \tag{3.1}$$

where $p(x, y)$ is the joint probability that the two words x and y co-occur in a given context, whereas $p(x)$ and $p(y)$ are the marginal probabilities. SVD is then applied to the PPMI matrix, which factorises $\mathbf{X}$ as $\mathbf{USV}^{\top}$, where $\mathbf{S}$ are the singular values of $\mathbf{X}$, $\mathbf{U}$ and $\mathbf{V}$ are orthonormal matrices of singular vectors[5]. $\mathbf{U}$ is truncated by keeping only the sub-matrix of the top d singular values to be the word embedding matrix $\mathbf{W} \in \mathbb{R}^{n \times d}$ (i.e., $\mathbf{W} = \mathbf{U}_d$). Consistent with Levy et al. (2015a), we empirically find that ignoring the singular matrix $\mathbf{S}$ when generating word embeddings performs better among the evaluated tasks and datasets. Following the literature, word embeddings with latent dimensions that are obtained after applying dimensionality reduction on the co-occurrence matrix are referred to as LSA (Deerwester et al., 1990).

As an alternative dimensionality reduction method, we also apply Nonnegative Matrix Factorisation (NMF) (Lee and Seung, 2001). Given the co-occurrence matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$, NMF computes the factorisation $\mathbf{X} = \mathbf{GH}$, where $\mathbf{G} \in \mathbb{R}^{n \times d}$, and $\mathbf{H} \in \mathbb{R}^{d \times m}$, and $\mathbf{G} \geq 0, \mathbf{H} \geq 0$ (i.e., $\mathbf{G}$ and $\mathbf{H}$ contain non-negative elements). By setting $d < \min(n, m)$, we

[5] sparsesvd package in python is used for SVD, `https://pypi.org/project/sparsesvd/`.

obtain lower d-dimensional embeddings for the rows and columns of $\mathbf{X}$, given respectively by the rows and columns in $\mathbf{G}$ and $\mathbf{H}$. Unlike SVD that generates dense embeddings of small positive or negative values on most of the dimensions across all word domains, the word embeddings obtained from NMF are interpretable because of the non-negative constraints and the sparsity as each word is represented by a small number of active dimensions (Murphy et al., 2012). By using non-negative sparse word embeddings, we can test the behaviour of word-level relational features from a different aspect. The pre-trained word embeddings on ukWaC for all the models are publicly available[6].

We experimented using both unnormalised and ℓ_2 normalised word embeddings and found that ℓ_2 normalised word embeddings perform better than the unnormalised version in most configurations of the conducted experiments in this thesis. Consequently, we report results obtained only with the ℓ_2 normalised word embeddings in the remainder of the thesis.

3.3 Relational Similarity: Datasets and Tasks

A natural way to gauge the effectiveness of relational features is to measure the similarity between two word-pairs. Given two word pairs (a, b) and (c, d), the task is to measure the similarity between the relations that exist between the two words in each pair. A good relational representation method assigns a high degree of relational similarity if (a, b) stands in the same relation as (c, d). This type of similarity is referred to as relational similarity and is analogous to attributional similarity, which is the correspondence between the attributes of two objects. Relational similarity measures can be involved in various relational tasks such as finding analogies, ranking word-pairs in a particular relation, completing proportional analogies of the form $a : b :: c :?$, and relation classification. In the following subsections, we lay out the tasks and the benchmark datasets evaluated throughout this thesis.

3.3.1 Multiple-Choice Analogy Questions: SAT

The Scholastic Aptitude Test (SAT) word analogy dataset contains 374 multiple choice questions that each includes a word-pair as the stem, and the examinees are required to select the most analogous word-pair out of four or five candidate answer word-pairs. SAT is introduced in Turney et al. (2003) as a way of evaluating methods for measuring the relational similarity. An example is shown in Table 3.1. Typically, we need to measure the similarity between the relation of the question word-pair and the relation of each of the candidate word-pairs to select the candidate with the highest relational similarity as the correct answer. The accuracy metric is used to report performance on the SAT, which is the ratio of the number of questions answered correctly to the total number of questions

[6] https://github.com/Huda-Hakami/Word-Embeddings-ukWaC

Table 3.1: An example question from the SAT dataset. In this question, the common relation between the stem (*ostrich*, *bird*) and the correct answer (*lion*, *cat*) is is-a-large.

Stem:		*ostrich* : *bird* ::
Choices:	(a)	*lion* : *cat*
	(b)	*goose* : *flock*
	(c)	*ewe* : *sheep*
	(d)	*cub* : *bear*
	(e)	*primate* : *monkey*
Solution:	(a)	*lion* : *cat*

in the dataset. Because there are five candidate answers out of which only one is correct, random guessing would provide 20% accuracy.

3.3.2 Ranking Word-Pairs: SemEval-2012 Task 2

Instances of relations can have different degrees of prototypicality. For instance, according to human ratings for word-pairs in PART-WHOLE (object: component) relation, "*hand*: *finger*" is assigned a higher degree to be an instance of the relation compared to "*computer*: *chip*", which in turn has a higher degree than "*movie*: *scene*". SemEval-2012 Task 2[7] is a benchmark dataset that is proposed for measuring the degrees of relational similarity to rank word-pairs according to the degree to which a relation applies (Jurgens et al., 2012). This dataset covers ten coarse-grained categories of semantic relations, each with several subcategories. The dataset includes a total of 79 fine-grained semantic relation types (10 for training and 69 for testing). Each relation type has approximately 41 word-pairs (not all are equally good examples for a relation) and three to four prototypical examples. In total, nearly $3,464$ word-pairs are collected for this dataset across relations. Table 3.2 illustrates some selected relation types along with the given ranked pairs in SemEval-2012 Task 2. The task assigns a score to each word-pair, which we compute by averaging the relational similarity between the given word-pair and prototypical word-pairs in a relation. Following previous work, the performance is evaluated by its correlation with human judgments using Spearman correlation across relations or the macro-averaged MaxDiff (Maximum Difference Scaling).

3.3.3 Analogy Completion: Google and MSR Analogies

An alternative task that involves measuring relational similarity is word analogy completion of the form: $a : b :: c :?$. The task is to find the missing fourth word d from a fixed vocabulary such that the relational similarity between (a, b) and (c, d) is maximised. For

[7] https://sites.google.com/site/semeval2012task2/

Table 3.2: Taxonomy of selected semantic relations in SemEval-2012 Task 2 with ranked word-pairs

Main Category	Subcategories	Prototypical pairs	Ranked pairs (highest to lowest)
Part-Wole	Object:Component	car:engin, face:nose	hand:finger ... toe:foot
	Mass:Portion	water:drop, time:moment	hour:seconds ... country:city
	Collection:Member	forest:tree, anthology:poem	army:soldiers ... album:songs
Class-Inclusion	Taxonomic	flower:tulip, poem:sonnet	weapon:spear ... insect:ant
	Functional	weapon:knife, ornament:brooch	tool:hammer ... appliance:fridge
	Class Individual	river:Nile, city:Berlin	ocean:pacific ... earth:planet
Cause-Purpose	Cause:Effect	enigma:puzzlement, joke:laughter	loss:grief ... run:sweat
	Case:Compensatory Action	hunger:eat, fatigue:sleep	thirst:drink ... dizzy:drunk
	Enabling Agent:Object	match:candle, gasoline:car	battery:flashlight ... match:wood

this task, we use the two datasets of MSR syntactic analogies (Mikolov et al., 2013c), and Google analogies[8] (Mikolov et al., 2013b). MSR analogies contains 8,000 proportional analogies covering ten different syntactic relations, such as "*highest* is to *high* as *worst* is to ?" where *bad* is the correct answer. The Google analogy benchmark contains 19,544 analogical questions covering nine syntactic and four semantic relation types, corresponding to 10,675 syntactic and 8,869 semantic analogies. The semantic questions are typically analogies about people or places, such as "*London* is to *England* as *Madrid* is to ?". We restrict the search space for the missing word to the words that appear in a large set of vocabulary consisting of 13,609 words, excluding the three words for each question. Following the literature, we report the accuracy of answering semantic (sem), syntactic (syn) questions separately, and also the total (sem and syn) accuracy.

3.3.4 Relation Classification: DiffVec

The DiffVec dataset was proposed by Vylomova et al. (2016), and consists of 12,458 triples (a, b, r), where word a and b are connected by a relation r. It is called DiffVec (Vector Differences) because it was initially proposed to evaluate vector differences over a large set of relation types. The relation set in the DiffVec comprises 15 coarse-grained relation types including lexical-semantic (e.g., Hypernym, Meronym and Causality), morphosyntactic paradigm (e.g., VerbPast and SingularPlural) and morphosemantic relations (e.g., CollectiveNoun and Light_Verb_Construction). Some of main relation types are classified into sub-categories, in total the dataset includes 36 fine-grained relations. Relation types along with word-pair examples for the DiffVec are shown in Table 3.3, and it is publicly available for evaluation[9].

In this thesis, we evaluate relational similarity on the DiffVec using a relation classification task. In relation classification, the problem is to classify a given pair of words (a, b) to a specific relation label r from a predefined set of relations according to the relation that

[8] `http://download.tensorflow.org/data/questions-words.txt`

[9] `https://github.com/ivri/DiffVec`

Table 3.3: Types of relations and the number of instances for each relation type in DiffVec.

Relation type	Sub-relations	Example	#Pairs
Hypernym	_	$(tool, knife)$	1,173
Meronym	_	$(tiger, mouth)$	2,825
Event	_	$(fix, oven)$	3,583
Collective-Noun	_	$(army, ants)$	2,57
Light-Verb-Construction	_	$(give, approval)$	58
Cause-Purpose	EnablingAgent: Object	$(battery, phone)$	34
	Cause: Effect	$(disease, sickness)$	38
	Agent: Goal	$(workers, salary)$	31
	Prevention	$(antibiotic, infection)$	33
	Instrument:Goal	$(aspirin, healing)$	29
	Instrument:IntendedAction	$(knife, cut)$	20
	Action:ActivityGoal	$(bath, clean)$	36
	Cause:CompensatoryAction	$(obesity, exercise)$	28
Space-Time	Item: Location	$(aquarium, fish)$	27
	Location-Process:Product	$(press, books)$	27
	Contiguity	$(frame, photograph)$	33
	Attachement	$(gloves, hand)$	27
	Sequence	$(inhale, smell)$	34
	LocationInstrument:AssociatItem	$(ball, sport)$	32
	LocationAction:Activity	$(kitchen, cooking)$	21
	Time-Action:Activity	$(morning, breakfast)$	34
Reference	Plan	$(map, city)$	35
	Sign: Significant	$(alarm, action)$	33
	Expression	$(crying, sadness)$	31
	Representation	$(song, emotion)$	30
	Knowledge	$(anatomy, body)$	27
	Concealment	$(encryption, data)$	31
Attribute	Object: TypicalAction(n.v)	$(dog, bark)$	33
	Object:State (n.n)	$(ice, cold)$	32
	Action:ObjectAttribute	$(collect, fee)$	6
Syntactic relations	Prefix	$(adjust, readjust)$	118
	Noun-SingPlur	$(artist, artists)$	100
	Verb_past	$(accept, accepted)$	100
	Verb_3rd	$(accept, accepts)$	99
	Verb_3rd_past	$(accepts, accepted)$	100
	VerbNoun_Nominalisation	$(abet, abetment)$	3,303
Total	36	_	12,458

exists between a and b. For the evaluation, we perform the 1-Nearest Neighbour (1-NN) classification with leave-one-out cross-validation. The testing set consists of a single word-pair, and the training includes the remaining word-pairs of a dataset. If the nearest neighbour has the same relation label as the target word-pair, then it is considered to be a correct classification. The micro-averaged classification accuracy is computed as the ratio of the correct matches to the total number of tested word-pairs. We avoid lexical overlaps between testing and training pairs. For example, given the test word-pair (a, b), we exclude the training pairs (a, c), (b, c), (c, a) or (c, b), if any.

Table 3.4: Relation types in BATS dataset.

Semantic Relations	Sub-relations	Example	Syntactic Relations	Sub-relations	Example
Lexicographic	hypernyms-animals	(*ant*, *insect*)	Inflectional	noun-plural-reg	(*car*, *cars*)
	hypernyms-misc	(*cake*, *dessert*)		noun-plural-irreg	(*academy*, *academies*)
	hyponyms-misc	(*weapon*, *gun*)		adj-comparative	(*cheap*, *cheaper*)
	meronyms-substance	(*jam*, *fruit*)		adj-superlative	(*huge*, *hugest*)
	meronyms-member	(*tree*, *forest*)		verb-inf-3pSg	(*accept*, *accepts*)
	meronyms-part	(*apartment*, *bedroom*)		verb-inf-Ving	(*add*, *adding*)
	synonyms-intensity	(*cat*, *lion*)		verb-inf-Ved	(*accept*, *accepted*)
	synonyms-exact	(*child*, *kid*)		verb-Ving-3pSg	(*adding*, *adds*)
	antonyms-gradable	(*big*, *small*)		verb-Ving-Ved	(*agreeing*, *agreed*)
	antonyms-binary	(*after*, *before*)		verb-3pSg-Ved	(*agrees*, *agreed*)
Encyclopedic	country-capital	(*Beijing*, *China*)	Derivational	noun+less-reg	(*arm*, *armless*)
	country-language	(*Cuba*, *Spanish*)		un+adj-reg	(*able*, *unable*)
	UK_city-county	(*Liverpool*, *Lancashire*)		adj+ly-reg	(*global*, *globally*)
	name-nationality	(*Hawking*, *British*)		over+adj-reg	(*excited*, *overexcited*)
	name-occupation	(*Edison*, *inventor*)		adj+ness-reg	(*aware*, *awareness*)
	animal-young	(*cat*, *kitten*)		re+verb-reg	(*adjust*, *readjust*)
	animal-sound	(*dog*, *bark*)		verb+able-reg	(*edit*, *editable*)
	animal-shelter	(*bee*, *hive*)		verb+er-irreg	(*bake*, *baker*)
	things-color	(*coal*, *black*)		verb+tion-irreg	(*accuse*, *accusation*)
	male-female	(*king*, *queen*)		verb+ment-irreg	(*agree*, *agreement*)
Total	20	1,000		20	1,000

3.3.5 Bigger Analogy Test Set

Bigger Analogy Test Set (BATS) is a dataset of word-pairs that is proposed by Gladkova et al. (2016), and it is publicly available[10]. BATS has relational instances that are classified into four main relation types, namely, lexicographic semantics, encyclopaedic semantics, inflectional morphology and derivational morphology. Each relation type is divided into ten subcategories of which each includes 50 word-pair examples. In total, BATS includes 40 relation types and 2,000 relational instances. Relation types along with word-pair examples taken from BATS are shown in Table 3.4. As with the DiffVec dataset, we use the relation classification task when evaluating on BATS.

3.4 Unsupervised Compositional Approaches for Relations

There is a considerable number of literature around representing large linguistic units, such as phrases and sentences from word-level representations, which is referred to as *compositional semantics* (Mitchell and Lapata, 2008; Baroni and Zamparelli, 2010; Socher et al., 2013b). However, the problem of representing the meaning of a sentence differs from the problem of representing the relation between two words, in several important ways. First, a sentence would often contain more than two words, whereas we consider word pairs that always contain precisely two words. Second, a good sentence representation must encode the meaning of the sentence in its entirety, ideally capturing the meanings of salient

[10] `http://vecto.space/projects/BATS/`

content words in the sentence. On the other hand, in relation representation, we are not interested in the meanings of individual words, but the relationship between two words in a word pair. For example, given the word pair (*ostrich*, *bird*), the semantics associated with *ostrich* or *bird* is not of interest to us. Instead, we want to represent the relation is-a-large that holds between the two words in this example. Most of the compositional operators that have been proposed in prior work on sentence representations, such as vector addition or element-wise multiplication, could be used to create relation representations for word-pairs, but there is no guarantee that the same operators that have been found to be effective for sentence representation will be accurate for relation representation. As we see later in this chapter, vector offset, which does not scale up to sentences turns out to be a better operator for relation representation.

To the best of our knowledge, there have been few systematic studies devoted to exploring the best compositional operator to be applied on word embeddings for relation representations. For example, the Gábor et al. (2017) study explores various vector and similarity combinations on a semantic space of word embeddings for measuring relational similarity. However, the authors do not represent relations in any way. Weeds et al. (2014) compare different combinations on word representations to train a classifier that distinguish hypernym and co-hypernym relation types. In response to this gap, a comprehensive study for unsupervised compositional operators that can be applied on pre-trained word embeddings to obtain word-pair representations is performed. The study presented in this section is limited to unsupervised functions that are nonparametric (i.e., they do not have learnable parameters), and are applied to word embeddings trained in an unsupervised fashion. Parametric functions that require training data for computing the optimal values of the parameters for composing relation representations from word embeddings are beyond the scope of this chapter.

The contributions made in this work can be summarised as follows. We conduct an empirical comparison of the compositional operators (offset, concatenation, addition and element-wise multiplication) to derive relational features between words from word embeddings. Following related work, these operators are called unsupervised methods for constructing relation representations. We investigate the performance of these operators by measuring the relational similarity between word-pairs for multiple relational tasks and benchmark datasets that were introduced in Section 3.3. We examine how the performance of different compositional operators are affected by the models used to obtain word embeddings and the dimensionality of such embeddings. The extent to which the PairDiff operator can encode relational directionality is also considered.

This section is organised as follows. Section 3.4.1 defines the compositional operators that are evaluated for relations between words. Next, experimental results for the introduced relational datasets are discussed in Section 3.4.2. Section 3.4.3 breakdowns the performance by relation types. The effect of the dimensionality of a word embedding space has been tested for relation learning as presented in Section 3.4.4. Section 3.4.5 presents the evaluation of relations directionality under PairDiff. In Section 3.4.6, we investigate the extent to which word embeddings learnt from a corpus can benefit relation predictions in KGs applying the unsupervised compositional operators.

3.4.1 Unsupervised Compositional Operators

Our goal is to compare different compositional operators for composing representations of the relation between two words given the corresponding word embeddings. We assume that pre-trained word embeddings are provided, and the task is to use these word embeddings to compose relation representations. Specifically, given a word-pair (a, b), consisting of two words a and b, represented respectively by their embeddings $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^d$, we evaluate different compositional operators/functions that return a vector $\boldsymbol{r} \in \mathbb{R}^\delta$ given by (3.2) representing the relationship between a and b.

$$\boldsymbol{r} = f(\boldsymbol{a}, \boldsymbol{b}) \tag{3.2}$$

We use the following operators to construct $\boldsymbol{r}$ for a given word-pair (a, b):

PairDiff: The pair difference operator has been used by Mikolov et al. (2013c) for detecting syntactic and semantic analogies using offset vectors. For example, given a pair of words (a, b), they argue that $(\boldsymbol{b} - \boldsymbol{a})$ produces a vector that captures the relation existing between the two words a and b. Under the PairDiff operator, a resultant relation representation vector has the same dimensionality as the input vectors. The PairDiff operator is defined as follows:

$$\boldsymbol{r} = (\boldsymbol{b} - \boldsymbol{a}) \tag{3.3}$$

PairDiff captures the information related to a semantic relation by the direction of the resultant vector. Similar relations have shown to produce parallel offset vectors in prior work on word embedding learnings (Pennington et al., 2014). Such geometric regularities are useful for NLP tasks, such as solving word analogies (Mikolov et al., 2013c; Levy and Goldberg, 2014).

Concat: The linear concatenation of two n-dimensional vectors $\boldsymbol{a} = (a_1, \ldots, a_n)^\top$ and $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)^\top$ produces a $2n$-dimensional vector $\boldsymbol{r}$ given by,

$$\boldsymbol{r} = (a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n)^\top.$$

Here, $\boldsymbol{r}$ can then be used as a proxy for the relationship between a and b. Vector concatenation retains the information that exists in both input vectors in the resulting composed vector. In particular, vector concatenation has been found to be effective for combining multiple source embeddings to a single meta embedding (Yin and Schütze, 2016). However, a disadvantage of concatenation is that it increases the dimensionality of the relation representation compared to that of the input word embeddings.

Mult: Apply element-wise multiplication between $\boldsymbol{a}$ and $\boldsymbol{b}$ such that the i^{th} dimension of $\boldsymbol{r}$ has the value of multiplying the i^{th} dimensions of the input vectors. Applying element-wise multiplication generates a vector in which the dimensions common to both words receive non-zero values. The Mult operator is defined as follows:

$$\begin{aligned}\boldsymbol{r} &= \boldsymbol{a} \odot \boldsymbol{b} \\ \boldsymbol{r}_i &= \boldsymbol{a}_i \boldsymbol{b}_i\end{aligned} \tag{3.4}$$

Element-wise multiplication has the effect of selecting the common dimensions to the embeddings of both words for representing their interrelationships. Prior work on compositional semantics showed that element-wise multiplication is an effective method for composing representations for larger lexical units, such as phrases or sentences from elementary lexical units such as words (Mitchell and Lapata, 2008). However, element-wise multiplication has an undesirable effect when the embeddings contain negative values. For example, two negative-valued dimensions can generate a positive-valued dimension in the relational representation. If the relations are directional (asymmetric), then such a change in sign can incorrectly indicate an opposite/reversed relations between words. For example, Baroni and Zamparelli (2010) report that word embeddings created via SVD perform poorly when composing phrase representations because of this sign-flipping issue. As we will see in Section 3.4.2, Mult also suffers from data sparseness because if at least one of the corresponding dimensions in two word embeddings is zero (or numerically close to zero), then the resultant dimension in the composed relational vector becomes zero. Our experimental results suggest that sparseness, more than negativity, is problematic for the Mult operator. However, to the best of our knowledge, the accuracy of element-wise multiplication has not been evaluated in the task of relation representation.

Add: Apply element-wise addition between $\boldsymbol{a}$ and $\boldsymbol{b}$ such that the i^{th} dimension of $\boldsymbol{r}$ has the value of adding the i^{th} dimensions of the input vectors, given as follows:

$$\begin{aligned}\boldsymbol{r} &= \boldsymbol{a} + \boldsymbol{b} \\ \boldsymbol{a}_i &= \boldsymbol{a}_i + \boldsymbol{b}_i\end{aligned} \tag{3.5}$$

Element-wise multiplication and addition have been evaluated in compositional semantics for composing phrase-level or sentence-level representations from word-level representations (Mitchell and Lapata, 2009, 2008). In the context of relations, a relationship might materialise between two entities because they share many common attributes. For example, two people might become friends through social media because they discover they have many common interests. Consequently, element-wise addition and multiplication emphasise such common attributes by adding their values together when composing the corresponding relation representation. In this work, we hypothesise that some relations are formed between entities because they have common attributes. By pairwise addition or multiplication of the attributes of two given words, we emphasise these common attributes in their relational representation.

Element-wise operators between word vectors assume that the dimensions of the word representation space are linearly independent. Alternatively, we can consider that the dimensions are cross-correlated and use cross-dimensional operators (i.e., operators that consider the i^{th} and j^{th} dimensions for $i = j$ as well as $i \neq j$) instead of element-wise operators to create relation representations. For this purpose, given a word representation matrix $\mathbf{W} \in \mathbb{R}^{n \times d}$ of n words and d dimensions, we create a correlation matrix $\mathbf{C} \in \mathbb{R}^{d \times d}$ in which the $\mathbf{C}_{ij}$ element is the Pearson correlation value of $\mathbf{W}_{:,i}$ and $\mathbf{W}_{:,j}$, (i.e., the i^{th} and the j^{th} dimensions for all of the represented words). In our preliminary experiments with the pre-trained word embeddings used as inputs, we found that the correlation coefficients between i, $j(\neq i)$ dimensions are close to zero, indicating that the dimensions are indeed uncorrelated (more details are in Chapter 4). Consequently, for the prediction-based word embeddings we used in this comparative study, we did not obtain any significant improvement in performance by using cross-dimensional operators. Therefore, we do not consider cross-dimensional operators for the purpose of relation representations.

3.4.2 Experiments and Results

We evaluate the effectiveness of the predefined operators for relations using the datasets along with the tasks that are shown in Section 3.3. For each operator f and a given word-pair (a, b), the relation vector $\boldsymbol{r}_{ab}$ is obtained by applying f on $\boldsymbol{a}$ and $\boldsymbol{b}$ embeddings. We adopt the cosine of the angle between two relation representations ($\boldsymbol{r}_{ab}$ and $\boldsymbol{r}_{cd}$) as a proxy of the relational similarity between the two pairs (a, b) and (c, d), which is defined as follows:

$$\mathrm{sim}(\boldsymbol{r}_{ab}, \boldsymbol{r}_{cd}) = \cos(\theta) = \frac{\boldsymbol{r}_{ab}{}^{\top} \boldsymbol{r}_{cd}}{||\boldsymbol{r}_{ab}||\, ||\boldsymbol{r}_{cd}||} \tag{3.6}$$

In Table 3.5, we compare the performance of the four compositional operators (PairDiff, Concat, Add and Mult) described in Section 3.4.1 for the word representation models described in Section 3.2. The best result for each dataset among the word embedding types is presented in bold. We observe that PairDiff achieves the best results compared with other operators for all the evaluated datasets and all word representation methods. PairDiff

Table 3.5: Accuracy (%) of the compositional operators for measuring relational similarity in the benchmark datasets. MaxDiff scores are reported for SemEval-2012 Task 2.

Representation model	Compositional operator	SAT	SemEval (MaxDiff)	MSR	Google			DiffVec	BATS
					sem	syn	total		
CBOW	PairDiff	**41.82**	**44.35**	**30.16**	**24.43**	**32.31**	**28.74**	**77.40**	**67.93**
	Concat	38.07	41.06	0.39	3.01	1.26	2.05	77.24	55.68
	Add	31.10	36.37	0.06	0.16	0.15	0.15	62.11	40.17
	Mult	27.88	35.19	8.13	2.38	6.11	4.42	61.01	36.47
SG	PairDiff	39.41	44.03	21.08	22.28	26.47	24.57	75.64	65.28
	Concat	35.92	41.21	0.30	1.40	1.17	1.27	73.03	49.42
	Add	28.69	35.48	0.00	0.17	0.13	0.15	55.92	36.12
	Mult	24.40	35.4	3.26	2.29	4.47	3.48	53.43	30.82
GloVe	PairDiff	41.02	42.8	16.74	15.42	21.00	18.47	72.87	60.63
	Concat	36.19	40.17	0.31	2.27	1.17	1.67	70.08	49.02
	Add	29.22	35.23	0.0	0.24	0.18	0.20	53.25	32.52
	Mult	23.32	32.0	0.91	3.87	1.39	2.51	41.4	22.06
LSA(SVD)	PairDiff	36.90	43.44	8.49	2.84	11.26	7.44	75.35	61.27
	Concat	38.77	42.04	0.35	0.50	0.82	0.68	72.48	47.59
	Add	31.82	36.05	0.01	0.26	0.14	0.19	54.55	36.03
	Mult	29.14	34.79	5.56	0.52	6.91	4.01	55.78	33.27
LSA(NMF)	PairDiff	35.29	42.88	2.8	1.75	3.66	2.79	70.08	51.94
	Concat	31.02	41.39	0.19	0.44	0.65	0.50	69.52	43.19
	Add	29.68	36.00	0.03	0.21	0.11	0.16	55.79	33.33
	Mult	21.12	34.49	0.00	0.03	0.00	0.02	44.32	21.71

reports significantly better results than Concat, Add and Mult for all the embedding types (both prediction- and counting-based) in the MSR, Google, DiffVec and BATS datasets according to Clopper-Pearson confidence intervals ($p < 0.05$). Because the SAT is the smallest dataset among all, so we were unable to see any significant differences with SAT.

Analogy completion in Google and MSR analogies are considered an open vocabulary task because to answer a question of the form "a is to b as c is to ?", we must consider all words in the corpus as candidates, which is an open vocabulary, not limited to the words that appear in the benchmark datasets, as in SAT or SemEval datasets. Therefore, applying PairDiff to each pair (a, b) and (c, d) retrieves candidates d that have relations with c similar to the relation between a and b, but not necessarily similar to the word c. For instance, the top three ranked candidates for the question "*man* is to *woman* as *king* is to ?" are *women*, *pregnant* and *maternity*. We notice that the top-ranked candidates indicate feminine entities. This observation explains the performance of PairDiff for answering MSR and Google analogies, which is lower compared with other relational tasks (similar observations have been made by Levy and Goldberg (2014)). Moreover, the open vocabulary task (Google and MSR analogies) is harder than the closed vocabulary task (SAT, SemEval, DiffVecs and BATS) because in closed vocabulary benchmarks we are provided with related word-pairs, whereas the number of incorrect candidates is much larger in the open vocabulary setting. This means that the probability of accidentally retrieving a noisy negative candidate as the

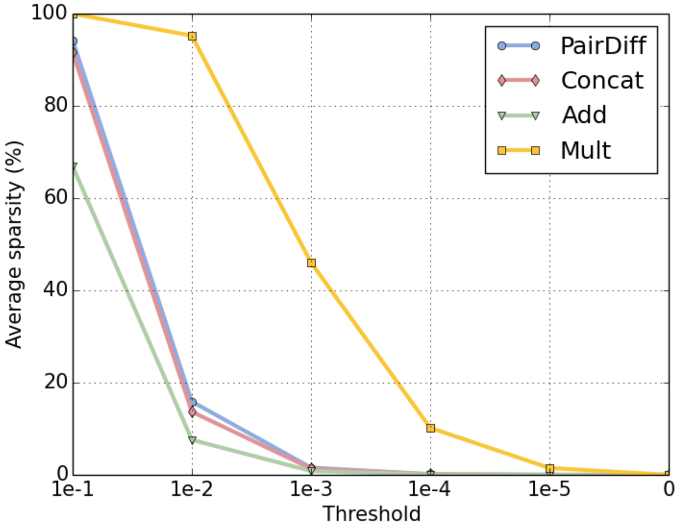


Figure 3.1: The average sparsity of relation embeddings for different operators using CBOW embeddings with 300 dimensions for some selected word-pairs.

correct answer is higher than in the closed vocabulary task. This property provides a reason for why Mikolov et al. (2013c) do not rely on the relation alone, and they consider shifting the context of $\boldsymbol{c}$ using a relation $\boldsymbol{r}$ when looking for $\boldsymbol{d}$, i.e., $\boldsymbol{r} + \boldsymbol{c}$. When using PairDiff, Mult and Add to construct $\boldsymbol{r}$ that is then used to shift $\boldsymbol{c}$, we respectively obtain 70.63, 28.02 and 5.35 on Google total analogies under the CBOW embeddings. Overall, we observe that Concat, Add and Mult fail in the task of analogy completion.

Mult is performing slightly worse with NMF word embeddings compared to other embedding models. Recall that NMF produces non-negative embeddings, and Mult conducts an elementwise multiplication operation on the two input word embeddings to create the embeddings for their relation. If the negativity was the only issue with the Mult operator, as previously suggested by Baroni and Zamparelli (2010), then Mult should have performed better with NMF. We hypothesise that the issue here is sparsity in the relation representations. To test the hypothesis empirically, we conduct the following experiment. First, we randomly select 140 word-pairs from the Google dataset and apply different compositional operators to create relation embeddings for each word-pair using 300-dimensional CBOW word embeddings as the input. Next, we measure the average sparsity of the set of relational embeddings created by each operator. We define sparsity at a particular cut-off level ϵ for a d dimensional vector as the percentage of elements with absolute value less than or equal to ϵ out of d. Formally, sparsity is given by (3.7).

$$\text{sparsity} = \frac{1}{d}\sum_{i=1}^{d}\mathcal{I}[|x_i|\leq \epsilon] \tag{3.7}$$

Here, $\mathcal{I}$ is the indicator function that returns 1 if the expression evaluated is true, or 0 otherwise. Our definition of sparsity is a generalisation of the ℓ_0 norm that counts the number of non-zero elements in a vector. However, exact zeros will be rare in practice, so we require a more sensitive measure of sparsity, such as the one given in (3.7). Average sparsity is computed by dividing the sum of sparsity values given by (3.7) for the set of word-pairs by the number of word-pairs in the set (i.e., 140).

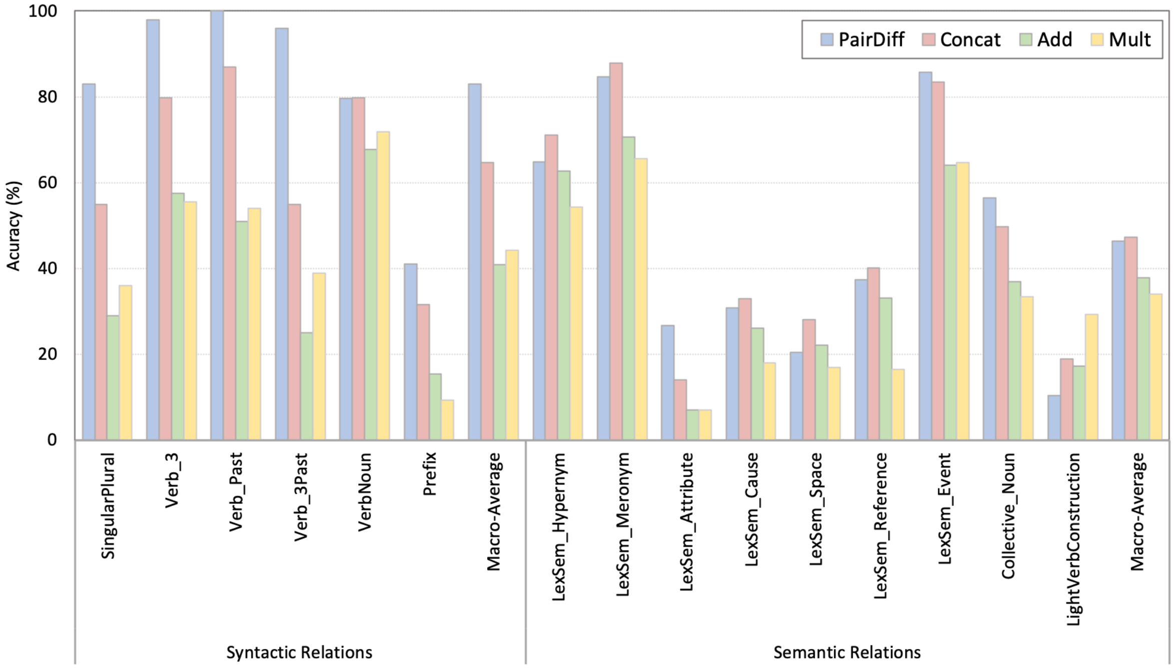
Figure 3.1 shows the average sparsity values for the operators under different ϵ levels. As shown in the figure, the Mult operator generates sparse vectors for relations compared to other operators under all ϵ values. Considering that Mult performs a conjunction over the two input word embeddings, even if at least one embedding has a nearly zero dimension, after element-wise multiplication we are likely to be left with nearly zero dimensions in the relation embedding. Such sparse representations become problematic when measuring cosine similarity between relation embeddings, which leads to poor performances in word analogy tasks.

3.4.3 Breakdown of the Performance by Relation Types

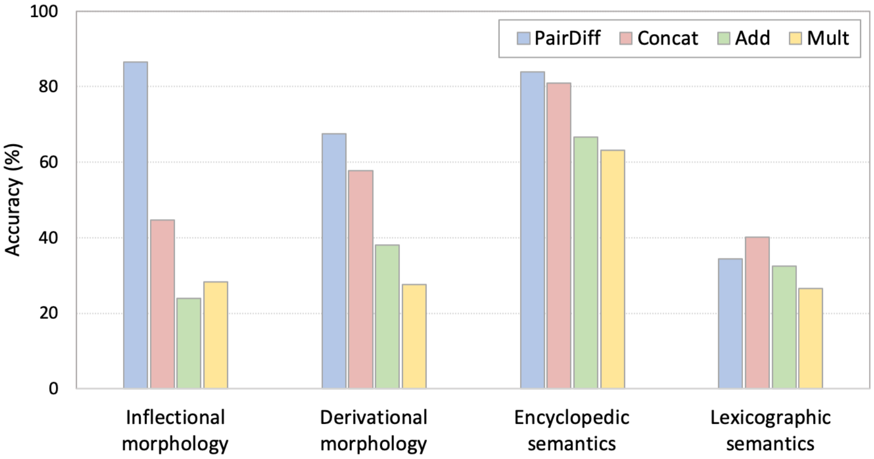
As elaborated in Section 3.4.2, the evaluation of relation embeddings generated by applying compositional operators on word embeddings reveals the optimality of the PairDiff operator for multiple tasks and datasets. However, it remains unclear how appropriate is the PairDiff for various relation types. To answer this question, we need to breakdown the evaluation by the relation types for a given benchmark dataset. Most of the relational benchmark datasets we introduced in Section 3.3 classify pairs of words to explicit linguistic relations, which enable us to compare the accuracy among different relation types. For this purpose, we use DiffVec and BATS that are evaluated for relation classification tasks and contain a variety of relation types, including semantic and syntactic relations.

Figure 3.2 reports the accuracy of each relation on DiffVec and BATS. Out of the nine semantic and morphosemantic relations in DiffVec, PairDiff performs better than other operators only for Attribute, Event and Collective-Noun relation types. For Hypernym, Meronym, Causality, SpaceTime and Reference, the concatenation of word embeddings achieves the best accuracy in relation classification. On the other hand, syntactic relations in DiffVec are encoded better as captured by PairDiff compared with other compositional operators, a finding that is consistent with previous works (Chen et al., 2017; Vylomova et al., 2016; Köper et al., 2015). Similarly, in the BATS dataset (Figure 3.2b), PairDiff fails to represent lexicographic semantic relations, and the best macro-average accuracy is reported for inflectional syntactic relation types followed by semantic encyclopedics.

Given that the PairDiff is an effective operator for deriving relational features of arbitrary relations, one would expect coherency between the offset vectors of word-pairs sharing a considered relation type. That means in the embedding space, the PairDiff vectors have to



(a) DiffVec



(b) BATS

Figure 3.2: Accuracy on different types of relations for DiffVec (a) and BATS (b) datasets.

be almost parallel for word-pairs related by the same relation. This hypothesis is tested by measuring the average of pair-wise cosine similarities between the PairDiff vectors in each relation type of BATS dataset. As shown in Figure 3.3, inflectional morphology and encyclopaedic relations have stronger PairDiff directions in the 300-dimensional CBOW embedding space compared to lexicographic semantics. The results in Figure 3.3 are

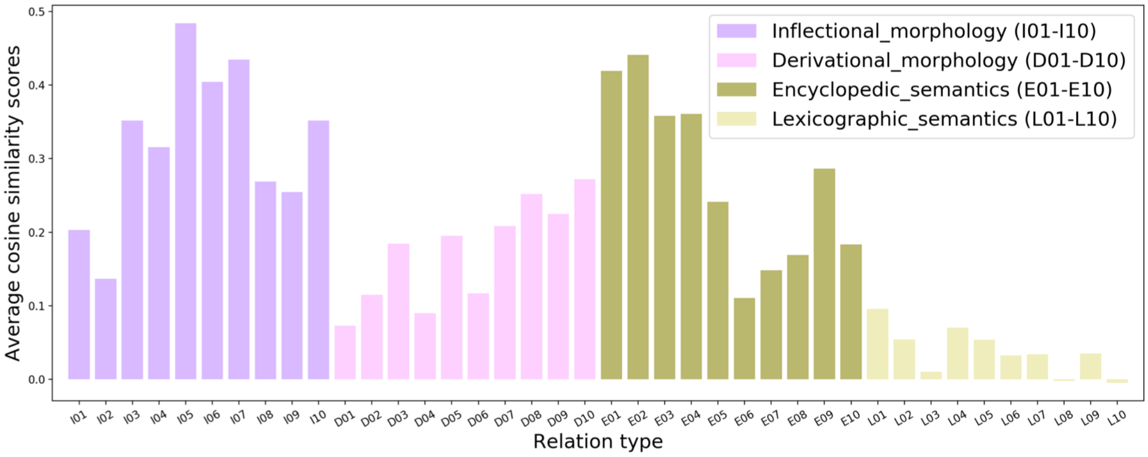


Figure 3.3: Average pair-wise cosine similarity scores for each relation type in BATS using 300-dimensional CBOW embeddings.

consistent with the accuracy of relation classification shown in Figure 3.2b, wherein the performance of PairDiff on lexicographic relations is poor. A possible explanation is that for encyclopaedic relations, the source words (i.e., a) can be grouped into a sub-space in the embedding space that is roughly aligned with the sub-space of the target words (i.e., b) (Liu et al., 2017; Bouraoui et al., 2018). For instance, in the country-capital relation the source words represent countries while the target words represent cities. On the other hand, lexicographic relation types do not have specific sub-spaces for the related head and tail words, which means that the offset vectors would not be sufficiently parallel for PairDiff to work well.

Let us analyse the role of PairDiff for morphological relation types to reason a relatively high accuracy of this type of relations compared to others. For example, the inflectional morphology relation (verb-infinitive, verb-ing) with the instance (*play*, *playing*). The contexts for *play* and *playing* are quite different, because *play* would occur with *like to*, *hate to*, etc. as prefixes, and *hockey*, *football*, etc. as suffixes (e.g., "*I hate/like to play hockey/football)*". However, the contexts for *playing* would be something like "*I like playing hockey*", "*playing hockey is my hobby*" and "*he is playing football*". Note that missing *to* before *playing* and missing *is* before *play* makes adequate contextual clues to learn embeddings that encode such syntactic properties[11]. Compared to syntactic relations, semantic relations are more expressive as many contextual patterns can be used to indicate a semantic relation between two words.

Even though we are expecting different morphological forms of a word (e.g., *play* and *playing*) to occur with similar contexts, the above-discussed syntactic contexts are still

[11] Thats why removing stopwords might by a bad idea when learning word embeddings.

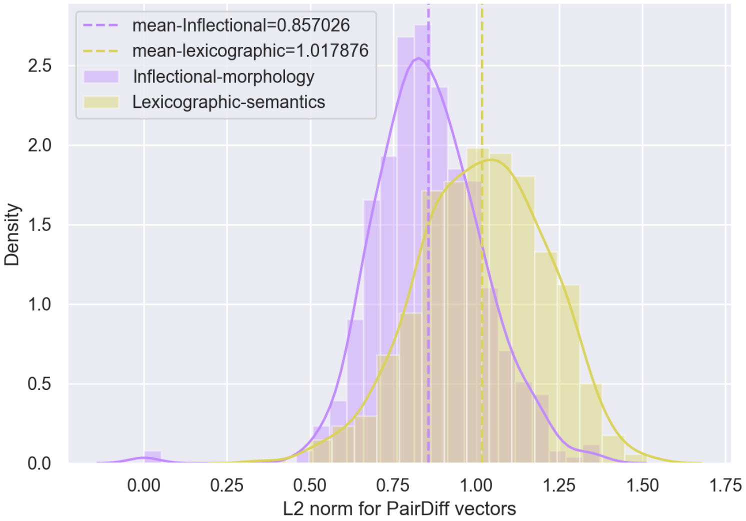


Figure 3.4: Probability density estimation for ℓ_2 norms of PairDiff vectors in inflectional morphology vs lexicographic semantic relations in BATS dataset.

encoded in their embeddings, which in turn explains the success of PairDiff method. We test this argument by measuring the ℓ_2 norm for PairDiff vectors of the ten inflectional morphology relations vs the ten lexicographic semantics in BATS. As shown in Figure 3.4, the distribution illustrates that PairDiff vectors of morphological word-pairs have relatively smaller norms than lexical semantics; however, PairDiffs between two different forms of words (i.e., as in *play* and *playing*) are still not close to zero.

Examining the compositional operators for each relation type shows the fact that the performance of the best operator (i.e., PairDiff) can vary from one relation type to another. Consequently, we conclude that a compositional operator for relations between words cannot capture all semantic relations in the space provided. This analysis also reveals that syntactic relations are predicted at high accuracy, whereas lexical semantic ones are more challenging. Thus, there exists much room for improvement.

3.4.4 Effects of the Word Embeddings Dimensionality on Relations

The dimensionality of the relational embeddings produced by the compositional operators depends on the dimensionality of the input word embeddings. For example, the Mult, Add, and PairDiff operators produce relational embeddings with the same dimensionality as the input word embeddings, whereas the Concat operator produces relational embeddings twice the dimensionality of the input word embedding. A natural question, therefore, is to consider how the performance of the relational embeddings varies with the dimensionality of the input word embedding. To study the relationship between the dimensionality of the input word embeddings and the composed relational embeddings, we conduct the following experiment. We first train word embeddings of different dimensionalities using the ukWaC corpus and keep all other parameters of the word embedding learning method fixed except for the dimensionality of the word embeddings learnt. Because CBOW turned out to be the single best word embedding learning method according to the results in Table 3.5, we use CBOW as the preferred word embedding learning method in this analysis. Figure 3.5 shows the performance of the compositional operators on the benchmark datasets using the CBOW input word embeddings with dimensionalities in the range of 50-800. As seen in the Figure, PairDiff outperforms all other operators across all dimensionalities. PairDiff reports the best results on SemEval and DiffVecs with 300 and 200 dimensions, respectively. Performance saturates when the dimensionality is increased beyond these points. On the other hand, SAT shows a different trend as the performance of PairDiff continuously increases with the dimensionality of the input word embeddings. We observe another behaviour from the figure for the MSR and Google datasets where the performance of PairDiff decreases while that of Mult increases with the dimensionality of the input word embedding.

To understand the above-described trends, we first note that the dimensions in word embeddings provide almost complementary information related to word semantics. As described in Section 3.4.1, correlations between different dimensions in the word embeddings are small, showing that different dimensions are uncorrelated. Adding more dimensions to the word embedding can be seen as a way of representing richer semantic information. However, increasing the dimensionality also increases the number of parameters to learn. Prediction-based word embedding learning methods first randomly initialise all the parameters and then update them such that the co-occurrences between words can be accurately predicted in a given contextual window. However, the training dataset, which in our case is the ukWaC corpus, is fixed. Therefore, we have more parameters than we can reliably estimate using the available data, resulting in overfitted noisy dimensions as we increase the dimensionality of the word embeddings learned.

One hypothesis for explaining the seemingly contradictory behaviour from the PairDiff and Mult operators in the open vocabulary task (MSR and Google analogy completion) is the following. When we increase the dimensionality of the input word embeddings, there will be some noisy dimensions in the input word embeddings. The PairDiff operator amplifies the noise in the sense that the resultant offset vector retains noisy high dimensions that appear in both word embeddings. On the other hand, the Mult operator behaves as a low-pass filter where we shutdown dimensions that have small (or zero) valued dimensions in at least one of the two embeddings via the element-wise multiplication of corresponding dimensions. Therefore, Mult will be robust against the noise that exists in the higher dimensions of the word embeddings compared to the PairDiff operator. To empirically test this hypothesis, the

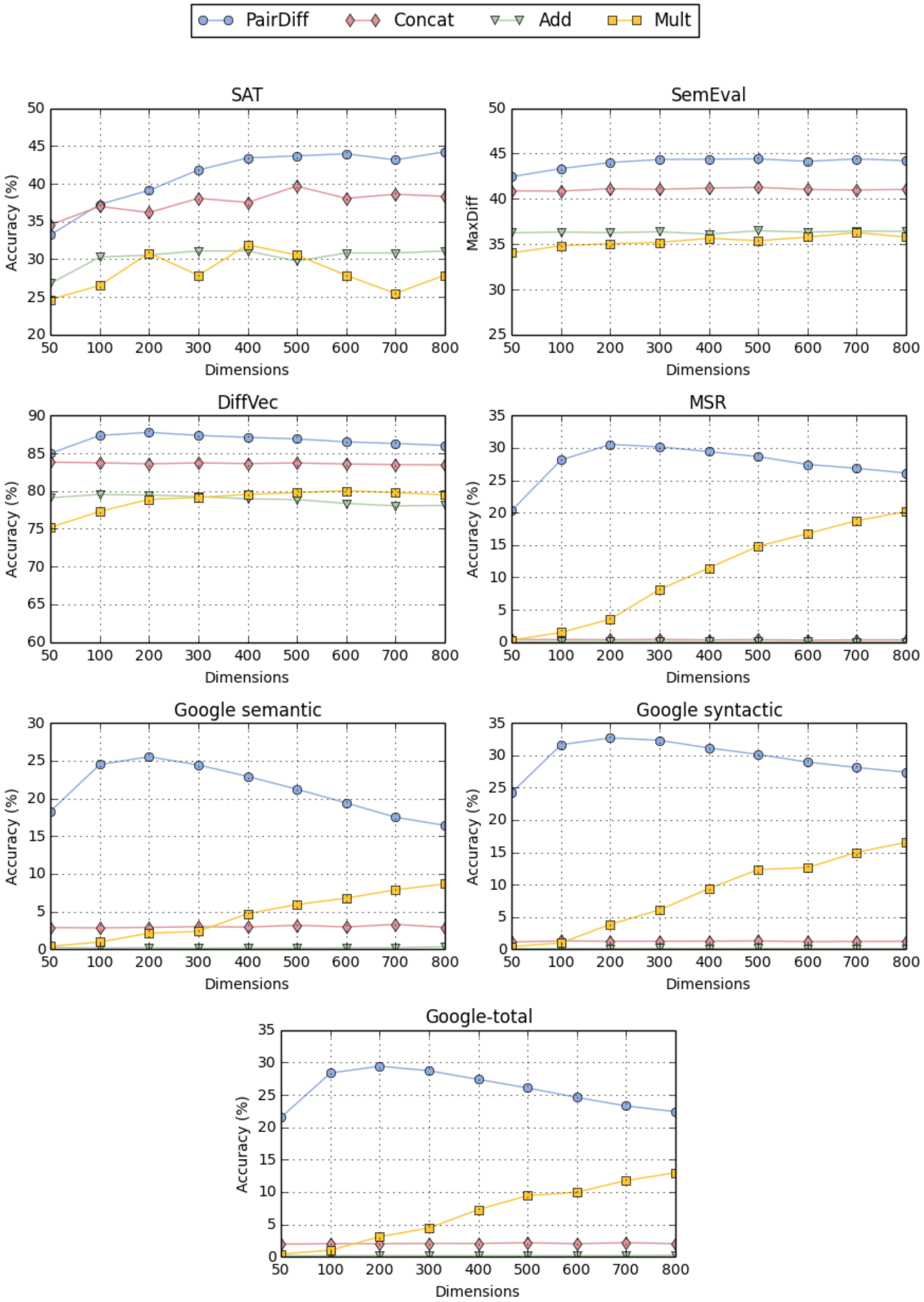


Figure 3.5: Influence of the dimensionality of the CBOW word embeddings for compositional relation representations.

ℓ_2 norms of $(\boldsymbol{a} - \boldsymbol{b})$ and $(\boldsymbol{a} \odot \boldsymbol{b})$ are computed for word embeddings of different dimensions and averaged over 140 randomly selected word-pairs. As shown in Figure 3.6, the norm of the PairDiff relation embeddings increases with dimensionality, whereas the norm of the relation embedding generated by Mult decreases. This result proves the hypothesis that Mult filters out the noise in high dimensional word embeddings better than PairDiff.

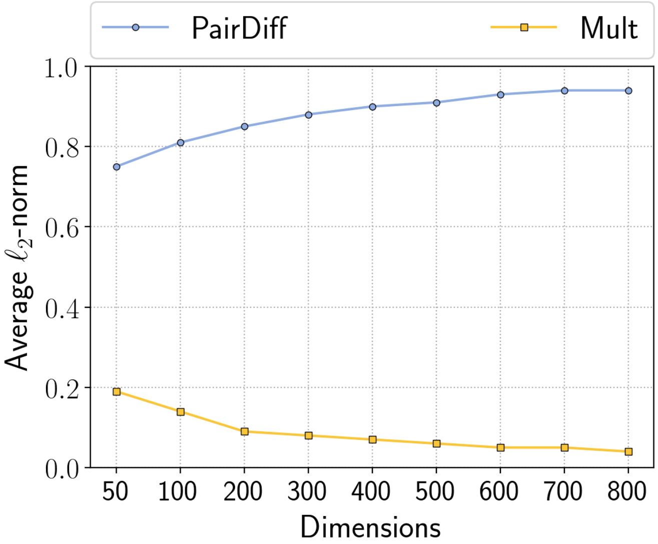


Figure 3.6: Average ℓ_2 norm of relational vectors generated using PairDiff and Mult operators.

3.4.5 Evaluating Relations Directionality Under PairDiff

Relations between words can be categorised as either being *symmetric* or *asymmetric*. If two words a and b are related by a symmetric relation r, then b is also related to a with the same relation r. Examples of symmetric relations include synonym and antonym. On the other hand, if a is related to b by an *asymmetric* relation, then b might not be necessarily related to a with that relation. Examples of asymmetric relations include hypernym and meronym. As discussed in Section 3.4.2, the PairDiff operator outperforms the Add and Mult operators in multiple relational tasks that involve measuring the relational similarity between word-pairs. Unlike Mult and Add, which are commutative operators, PairDiff is a non-commutative operator. This fact raises the question of whether PairDiff can detect the directionality of relations between words.

To test the ability of PairDiff for detecting the direction of a relation, we set up the following experiment. Using a set of word-pairs with a common directional relation r between the two words in each word pair as the training data, we use PairDiff to represent the relationship between two words in a word-pair, given the word embeddings for those two words. Next, we swap the two words in each word-pair and apply the same procedure

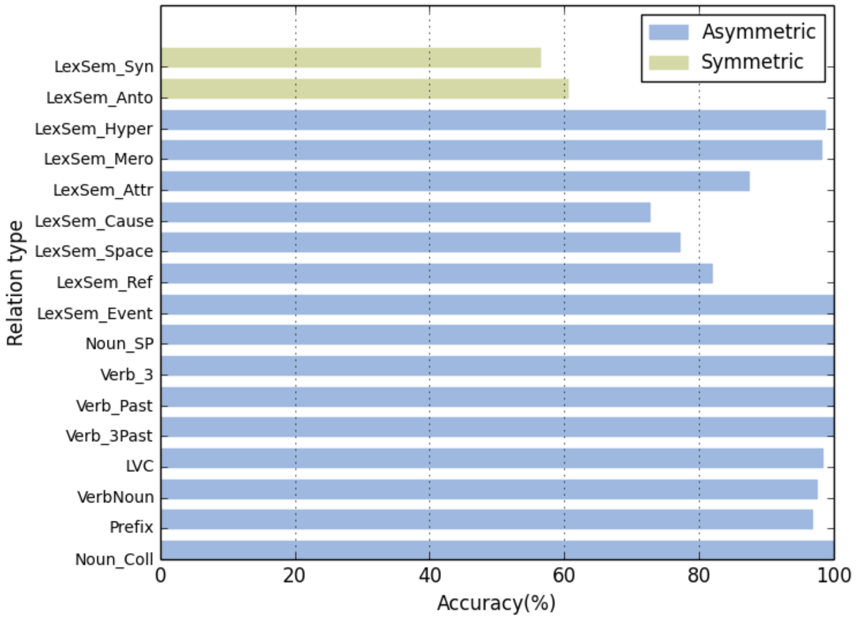


Figure 3.7: The accuracy of SVM classifier for evaluating the directionality of relation embeddings using PairDiff.

to create relation embeddings for the reversed relation r' in each word-pair. We model the task of predicting whether a given word-pair contains the original relation r or its reversed version r' as a binary classification task. Specifically, we train a binary Support Vector Machine (SVM) with a linear kernel with the cost parameter set to 1 using held-out data. If the trained binary classifier correctly predicts the direction of a relation in a word-pair, then we conclude that the relation embedding for that word-pair accurately captures the information about the direction of the relationship that exists between the two words in the word-pair. We repeat this experiment with symmetric and asymmetric relation types and compare the performances of the trained classifiers to understand how well the directionality in asymmetric relations is preserved in the PairDiff embeddings.

For the asymmetric relation types, we use all relation types in the DiffVec because this dataset contains only these types of relations. For symmetric relation types, we use two popular symmetric semantic relations, namely, synonymy[12] and antonymy[13]. We report the five-fold cross-validation accuracies with each relation type, as shown in Figure 3.7. If the classifier reports a high classification accuracy for asymmetric relations compared to symmetric relations, then it indicates that the relation embedding can encode the directional information in a relation. From the Figure, we see that the accuracies for the two symmetric

[12] http://saifmohammad.com/WebDocs/LC-data/syns.txt
[13] http://saifmohammad.com/WebDocs/LC-data/opps.txt

relation types are lower compared to the asymmetric relation types. This result indicates that PairDiff correctly detects the direction in the asymmetric relation types.

3.4.6 Knowledge Graph Completion

So far, the focus of the evaluation has been on measuring the relational similarity between word-pairs to conduct various relational tasks. However, we still do not know which composition method to choose for other NLP tasks like KG completion. For this purpose, we analyse such compositional operators in the context of KG link predictions (i.e., KG completion). Specifically, we want to study the extent to which pre-trained word embeddings capture relational attributes for entities in a KG, and which unsupervised operator can perform well on predicting missing links.

Experimental Settings

KGs such as WordNet and Freebase link entities according to numerous relation types that hold between entities. Automatic KG completion attempts to overcome the incompleteness of such KGs by predicting missing relations in a KG. For instance, given a first entity (a.k.a. the head entity h) and a relation type r, we need to predict a second entity (a.k.a. the tail entity t) such that h and t are related by r.

To evaluate the unsupervised compositional operators for the KB completion task using the embeddings of KG entities from a text corpus, we apply the following procedure. To avoid the need for composing word embeddings to construct representations for multiple words entities, we used the WN18RR dataset (a subset of WordNet released by Dettmers et al. (2018)) because it primarily includes unigram entities[14]. In this experiments, we exclude entities consisting of more than one word. To evaluate the accuracy of a relation composition operator f, we first create a representation $\boldsymbol{r}_i$ for each relation type r_i using the entity pairs (h, t) in the training data by applying f to the embeddings of the two entities h and t as follows:

$$\boldsymbol{r}_i = \frac{1}{|\mathcal{T}_i|} \sum_{(h,r_i,t)\in\mathcal{T}_i} f(\boldsymbol{h}, \boldsymbol{t}) \tag{3.8}$$

Here, $\mathcal{T}_i$ is the set of pairs of entities that are related by r_i. Next, for each test triple $(h', r'_i, ?\,)$, we predict a distribution for the missing tail t' as follows:

$$\boldsymbol{t}' = \boldsymbol{h}' + \boldsymbol{r}'_i \tag{3.9}$$

We rank all the entities in WN18RR according to the cosine similarity score between the corresponding entity embedding with the predicted tail embedding. A similar procedure of

[14] Textual Information about WN18RR entities is taken from: `https://github.com/villmow/datasets_knowledge_embedding/tree/master/WN18RR`

Table 3.6: Results of the compositional operators for KG completion task.

	CBOW		SG		GloVe	
Method	MR	H@10	MR	H@10	MR	H@10
PairDiff	**1,580**	**25.72**	**1,393**	**27.18**	**1,806**	**25.39**
Add	1,831	23.93	1,550	25.62	2,154	15.97
Mult	1,814	24.63	1,641	25.62	2,088	23.41
BL	1,814	24.59	1,640	25.67	2,043	23.46

evaluating word embeddings for KG predictions has been used by Gupta et al. (2017).

If the correct tail entity can be accurately predicted in the top of the ranked list using the relation embeddings created by applying a particular compositional operator, then we can conclude that operator to be accurately capturing the relational information. Two measures have been used for evaluating the predicted tail entities: Mean Rank (MR) and Hits@10. MR is the average rank assigned to the correct tail entity in the ranked listed of candidate entities. A lower MR is better because the correct candidate is ranked at the top by the compositional operator under evaluation. Hits@10 is the proportion of correct entities that have been ranked among the top 10 candidates. It is noteworthy that our purpose here is not to propose state-of-the-art KG completion methods, but rather to use KG completion only as an evaluation task to compare different compositional operators for relation prediction. Prior work in KG completion learns entity and relation embeddings that can accurately predict the missing relations in a KG as described in Section 2.7.2.

Results

We exclude the Concat operator because according to the adopted evaluation in (3.9), addition is not well-defined between two vectors of different dimensionalities (i.e., we can pad up vectors with zeros, but dimensions will not correspond, which is an important issue). We also include a baseline (BL) in which the relation vector $\boldsymbol{r}'_i$ in (3.9) is ignored, and tail entities are ranked based on the similarity to the given head embedding. In total, we evaluated 11 relations in WN18RR, 19,144 unigrams entities in which we have their embeddings, $57,280$ training and $2,123$ testing triples across the relations. We use 300-dimensional prediction-based embeddings explained in Section 3.2.

Table 3.6 displays the performance on the compositional operators for the KG completion task on WN18RR, where low MR and high Hits@10 indicates better performance. As can be seen from the Table, the PairDiff operator yields the lowest MR and the highest Hits@10 accuracy among other operators for the three word embedding models. Overall, SG performs the best compared to CBOW and GloVe embedding models. Add and Mult similarly perform as BL that ignore the relation vector from the training triples. If a relation is asymmetric such as hypernym and has-part as in WN18RR, the addition model will be insensitive to the directionality of such relations compared to PairDiff which explains the better performance of PairDiff over Add and Mult. Even though the proposed results are far from those obtained by using KGE methods, it would be possible to bridge the embeddings obtained from a text corpus and KGEs for complementarity of corpus- and KG-based embeddings.

3.5 Representative Space for Relational Similarity

In the previous Section (3.4), we investigated unsupervised compositional operators that can be applied to word embedding space to encode relational features between words. Subsequently, relational similarity can be measured between two relation representations of the corresponding word-pairs (a, b) and (c, d). On the other hand, the relational similarity can be inferred from the similarity of the corresponding relation arguments (i.e., between a and c, b and d). Thus, the relational similarity score between (a, b) and (c, d) can be defined as a function of the two pairs that can be decomposable as similarities between a and c, b and d. For example, it is highly probable that (*electricity*, *wire*) and (*water*, *pipe*) are relationally similar because *electricity* and *water* share multiple properties such as they can flow, whereas *wire* and *pipe* are similar since they are both carrying things that flow. Thus, we can consider the two pairs as instances of the same relation, namely *flows in*.

This section focuses on the task of measuring the relational similarity between two word-pairs considering the argument-wise similarity using word representation features. In the previous section, prediction- and counting-based embeddings are employed to evaluate compositional operators for accessing relations between words. In contrast to the latent features in prediction-based embeddings, the counting-based approach represents words in terms of the observed (i.e., interpretable) co-occurring contextual features from a corpus. However, the features that accurately express the relational similarity between word-pairs from contextual features of individual words remain largely unknown. Previous studies proposed solutions based on linguistic intuitions such as *domain* (i.e., topic) and *function* (i.e., role) spaces, which consist respectively of nouns and verbs to represent words (Turney, 2012). The intuition behind the dual-space model is that the domain of a word is better described linguistically using nouns that appear around it, whereas the function of words can be expressed by surrounding verbs. Measuring similarities between words under this duality can help to infer relationally similar word-pairs. For example, for the analogy *electricity* : *wire* :: *water* : *pipe*, *electricity* and *wire* are from the domain of *electronic* that can be defined by nouns such as *energy*, *charge*, *power* and *circuit*. On the other hand, *electricity* and *water* sharing the same role in the function space as they occur with similar verbs such as *flow*, *runoff*, *get in* and *come on*.

Although the above-mentioned linguistically-oriented spaces for semantic relations are justified by experiments, the question *whether we can learn descriptors of semantic relations from labeled data?* remains unanswered. We address this question by proposing a method for ranking lexical features to represent the semantic relations existing between two words. Given a set of word-pairs labelled by their relation types, we model the problem of extracting descriptive features for relations as a linear classification problem. Specifically, a linear-SVM classifier is trained to discriminate between positive (analogous) and randomly generated pseudo-negative (non-analogous) word-pairs using statistical co-occurrence features associated with individual words. The weights learnt by the classifier for the features can then be used as a ranking score for selecting the most representative features for semantic relations. Experimental results on a benchmark dataset for relation classification show that the proposed feature selection method outperforms several competitive baselines and the previously proposed heuristics by Turney (2012). It is worth noting that selecting features using classification-based approaches has been adopted for different NLP tasks such as sentiment analysis (Tripathi and Naganna, 2015) and text classification (Mladenić et al., 2004; Chang and Lin, 2008).

This section is organised as follows. Section 3.5.1 discusses the task of measuring relational similarity by considering the word representation feature space. The proposed method for weighting word representation features that are discriminators for semantic relations is presented in Section 3.5.2. Section 3.5.3 demonstrates the experimental settings and results obtained in this study.

3.5.1 Relational Similarity in Feature Space

Let us consider a feature x in some feature space $\mathcal{S}$. No constraints have been imposed on the type of features here, and the proposed method can handle any type of features that can be used to represent a word such as other words that co-occur with a target word in the corpus (lexical features), or their syntactic categories such as Parts-Of-Speech (POS) (syntactic features). The feature space $\mathcal{S}$ is defined as the set containing all features we extract for all target words. The salience of x in $\mathcal{S}$ is represented by the discriminative weight $w(x, \mathcal{S}) \in \mathbb{R}$. For example, if x is a representative feature of $\mathcal{S}$, then it will have a high $w(x, \mathcal{S})$. The concept of a discriminative weight can be seen as a feature selection method. If a particular feature is not a good representative of the space, then it will receive a small (ideally zero) weight, thereby effectively pruning out the feature from the space.

Given the above scenario, the task of discovering relational feature spaces can be modelled as a problem of computing the discriminative weights for features. We use $\phi(a)$ to denote the set of non-zero features that co-occur with the word a. The salience $f(a, x, \mathcal{S})$ of x as a feature of a in $\mathcal{S}$ is defined as follows:

$$f(a, x, \mathcal{S}) = h(a, x) \times w(x, \mathcal{S}) \tag{3.10}$$

Here, $h(a, x) \geq 0$ is the strength of association between a and x, and can be computed using any non-negative feature co-occurrence measure. In the experiments, we use PPMI, defined in (3.1), computed using corpus counts as $h(a, x)$.

Equation (3.10) is analogous to the tf-idf score used in information retrieval in the sense that $h(a, x)$ corresponds to the term-frequency (tf) (i.e., how significant is the presence of x as a feature in a), and $w(x, \mathcal{S})$ corresponds to the document-frequency (df) (i.e., what is the importance of x as a feature in the space $\mathcal{S}$). The similarity, $\text{sim}_{\mathcal{S}}(a, c)$ between two words a and c in $\mathcal{S}$ can then be defined as in (3.11), which is the sum of pointwise products over the intersection of the feature sets $\phi(a)$ and $\phi(c)$.

$$\text{sim}_{\mathcal{S}}(a, c) = \sum_{x \in \phi(a) \cap \phi(c)} f(a, x, \mathcal{S}) f(c, x, \mathcal{S}) \tag{3.11}$$

Moreover, by substituting (3.10) in (3.11) we get:

$$\text{sim}_{\mathcal{S}}(a, c) = \sum_{x \in \phi(a) \cap \phi(c)} h(a, x) h(c, x) w(x, \mathcal{S})^2 \tag{3.12}$$

Following the proposal by Turney (2012), we can then compute the relational similarity, between two word-pairs (a, b) and (c, d) as the geometric mean of their functional similarities as follows:

$$\text{sim}_{\text{rel}}((a, b), (c, d)) = \sqrt{\text{sim}_{\mathcal{S}}(a, c) \times \text{sim}_{\mathcal{S}}(b, d)} \tag{3.13}$$

3.5.2 Learning the Relational Feature Space

The relational similarity measure described in Section 3.5.1 depends on the feature space $\mathcal{S}$ via the discriminative weights $w(x, \mathcal{S})$ assigned to each feature x. Therefore, our goal of discovering a representative feature space from the data is solved through the learning $w(x, \mathcal{S})$. We propose a supervised classification-based approach for computing the discriminative weights using a labelled dataset.

Let us define a labelled dataset as consisting of the word-pairs (a, b) and (c, d) annotated for $l = 1$ (i.e., the two word pairs are analogous) or $l = 0$ (otherwise). Here, $l \in \{0, 1\}$ denotes the class label. From (3.13) and (3.12), we see that for two analogous word-pairs, (a, b) and (c, d), their relational similarity increases if the two products $h(a, x)h(c, x)$ and $h(b, x)h(d, x)$ increase. Following this observation, we define a feature x to appear in an instance of word-pairs (a, b) and (c, d) if and only if:

$$(x \in \phi(a) \cap \phi(c)) \vee (x \in \phi(b) \cap \phi(d)) \tag{3.14}$$

Linear Classifier for Relational Feature Ranking

For the proposed classification-based approach, each positive instance word pairs $((a, b), (c, d))$ or negative word-pairs $((a', b'), (c', d'))$ has a corresponding feature vector in $\mathcal{S}$, such that the entry for x in the $(a, b), (c, d)$ positive instance is defined as follows:

$$g\left(\left(\left(a, b\right), \left(c, d\right)\right), x\right) = \mathcal{I}\left[x \in \phi(a) \cap \phi(c)\right] + \mathcal{I}\left[x \in \phi(b) \cap \phi(d)\right] \tag{3.15}$$

Here, $g(((a, b), (c, d)), x)$ denotes the value of the feature x in the feature vector representing the instance $((a, b), (c, d))$, and $\mathcal{I}$ is the indicator function that returns 1 if the expression evaluated is true, or 0 otherwise, and likewise for a negative instance. We train a linear-SVM binary classifier to learn a weight for each feature in the feature space. The function $w(x, \mathcal{S})$ can be interpreted as the confidence of the feature as an indicator of the strength of analogy (relational similarity) between (a, b) and (c, d). The absolute value of the weight of a feature can be considered as a measure of the importance of that feature when discriminating the two classes in a binary linear classifier. Therefore, we rank the features in the space according to the absolute value of the weights $|w(x, \mathcal{S})|$. Only the linearised kernel classifier explicitly associates the weights to individual features. Therefore, this approach is restricted to the linear kernel. In the case of non-linear kernels, such as polynomial kernels that can be expanded prior to learning all feature combinations considered in the kernel computation, we can still apply this technique to identify salient feature combinations. However, we limit the discussion in this work to finding relational feature spaces consisting of individual features and defer the study of salient feature combinations for relational similarity measurement to future work.

The proposed method is compared to Kullback–Leibler divergence (KL), PMI, heuristic verb space and random selection. The KL and PMI methods also require labelled data as in our proposed classification-based approach, as will be discussed in the following sections.

KL Divergence-based Ranking

For sentence-level similarity, Ji and Eisenstein (2013) apply a data-driven approach for weighting the features in the paraphrase classification task. Based on labelled data, they proposed a new weighting metric to distinguish the deterministic features for sentence semantics. The metric uses KL Divergence to weight the distributional features in the co-occurrence matrix for sentences before the decomposing process. They report significant improvement in sentence similarity in comparison with other works.

Inspired by the above-mentioned study, we evaluate the proposed classification-based approach against the KL divergence-based weighting approach to compute $w(x, \mathcal{S})$ for the relational similarity measurement. We consider the two distributions for each feature x in S-space, namely, $p(x)$ and $q(x)$ where $p(x)$ is computed for analogous pairs $((a, b), (c, d))$, while $q(x)$ is taken over the unrelated pairs of words $((a', b'), (c', d'))$. The two probability distributions are formally defined by (3.16).

$$\begin{aligned} p(x) &= P(x \in \phi(a) \mid x \in \phi(c), l = 1 \text{ or } x \in \phi(b) \mid x \in \phi(d), l = 1), \\ q(x) &= P(x \in \phi(a') \mid x \in \phi(c'), l = 0 \text{ or } x \in \phi(b') \mid x \in \phi(d'), l = 0) \end{aligned} \tag{3.16}$$

Specifically, we compute the probability $p(x)$ of a feature x being an indicator of the analogous class as follows:

$$\frac{1}{Z_p(x)} \sum_{(a,b),(c,d) \in \mathcal{D}_+} g\left(\left((a, b), (c, d)\right), x\right) \tag{3.17}$$

Here, $\mathcal{D}_+$ is the set of positive word-pairs, and the normalisation coefficient $Z_p(x)$ satisfies, $\sum_{x \in \mathcal{S}} p(x) = 1$. Likewise, we can compute $q(x)$, the probability of a feature x being an indicator of the negative (relationally dissimilar) class using the features occurrences in negative instances $((a', b'), (c', d'))$ as follows:

$$\frac{1}{Z_q(x)} \sum_{(a',b'),(c',d') \in \mathcal{D}_-} g\left(\left((a', b'), (c', d')\right), x\right) \tag{3.18}$$

Here, $\mathcal{D}_-$ is the set of negative word-pairs, and the normalisation coefficient $Z_q(x)$ satisfies, $\sum_{x \in \mathcal{S}} q(x) = 1$. Having computed $p(x)$ and $q(x)$, we then compute $w(x, \mathcal{S})$ as the KL divergence between the two distributions as,

$$w(x, \mathcal{S}) = p(x) \log \left(\frac{p(x)}{q(x)} \right). \tag{3.19}$$

PMI-based Ranking

The PMI-based approach to select a subset of informative features uses a mutual information-based methodology. The PMI statistical weighting method has been applied for feature selection in document categorisation (Xu et al., 2007; Schneider, 2005). It calculates the amount of information that a feature includes about a specific category. Xu et al. (2007) show that PMI is not an efficient approach to select relevant features for text classification compared with other known approaches, such as Document Frequency and Information Gain.

In this study, PMI is used to weight a feature x such that:

$$w(x, \mathcal{S}) = \text{PMI}(x, \mathcal{D}_+) - \text{PMI}(x, \mathcal{D}_-) \tag{3.20}$$

where $\text{PMI}(x, \mathcal{D}_+)$ measures the association between a feature x with analogues word-pairs, and $\text{PMI}(x, \mathcal{D}_-)$ indicates the co-occurrence of a feature with relationally dissimilar pairs. PMI is computed as follows:

$$\begin{gathered}\text{PMI}(x, \mathcal{D}_+) = \log\left(\frac{h(x, \mathcal{D}_+)}{h(x, \mathcal{D})|\mathcal{D}_+|}|\mathcal{D}|\right) \\ \mathcal{D} = \mathcal{D}_+ \cup \mathcal{D}_-\end{gathered} \tag{3.21}$$

Here, $\mathcal{D}$ is the union set of the positive and negative word-pairs and $h(x, \mathcal{D}_+)$ is summed for all analogous pairs as: $\sum_{(a,b),(c,d) \in \mathcal{D}_+} g\left(\left((a, b), (c, d)\right), x\right)$. Similarly, $h(x, \mathcal{D}_-)$ is calculated by considering the negative instances in the dataset.

We rank the features according to the absolute values of their weights by each of these methods to define the representative space to measure the relational similarity. The relational similarity between two given word pairs is computed as defined in (3.13) after reducing the word representations to the top-ranked feature space. We experimented using both unnormalised and ℓ_2 normalised word representations. We found that the ℓ_2 normalised word representations perform better than the unnormalised version in most configurations. Consequently, we report results obtained only with the ℓ_2 normalised word representations.

3.5.3 Experiments and Results

Dataset. The above-mentioned feature selection methods require a dataset of word-pairs labelled by their relation types to generate analogous and non-analogous relational instances. We use the following procedure leveraging the DiffVec dataset introduced in Section 3.3.4. Recall, DiffVec consists of triples (a, b, r) where word a and b are connected by a relation r. This dataset consists of 15 relation types; however, we include the relation types with an adequate number of pairs to generate the dataset. Consequently, seven semantic relation types and their subcategories are considered in this study, as listed in Table 3.7. For each relation, we exclude some pairs of words for testing the methods; in total, we have 367 testing pairs distributed among the relations. We generate positive training instances by pairing word-pairs that have the same relation type (considering sub-relations), resulting in $7,187$ positive instances from this procedure. Next, we randomly pair a word-pair from a relation r with a word-pair from a relation r', such that $r \neq r'$ to create a pseudo-negative training dataset with approximately an equal number of instances as that in the positive training dataset (i.e., $7,000$).

Table 3.7: Statistic of the dataset used to discover relational feature space.

Relation type	#Positive training	#Testing instances
Hypernym	1,100	57
Meronym	1,100	57
Event (objects action)	1,100	57
Cause-Purpose	1,149	56
Space-Time	1,435	56
Reference	1,047	54
Attribute	256	30
Total	7,187	367

Evaluation Settings

During the evaluation, we consider the problem of classifying a given pair of words (a, b) to a specific relation r in a predefined set of relations $\mathcal{R}$ according to the relation that exists between a and b. We measure the relational similarity between a given pair and all the remaining pairs in the testing data. Then, we perform 1-NN relation classification such that if the 1-NN has the same relation label as the target pair, then we consider it a correct match. Macro-averaged classification accuracy is used as the evaluation measure. We use the PPMI matrix generated by Turney et al. (2011) that contains PPMI values between a word and unigrams from the left and right contexts of that word in a corpus[15]. The total number of features extracted ($|\mathcal{S}|$) is $139,246$.

For a classification method, we train a linear SVM using scikit-learn library[16]. We use five-folds cross-validation to find the optimal value of the penalty parameter C of the error term. Following Turney (2012), we use verbs as $\mathcal{S}$ to evaluate the performance of the functional space for measuring relational similarity. We use the NLTK POS tagger[17]for identifying verbs in the feature space, and the verb space that is identified by the POS tagger contains $12,000$ verbs.

Results

In Table 3.8, we compare the feature weighting methods discussed in Section 3.5.2 for different semantic relation types used in the evaluated dataset (illustrated in Table 3.7). The accuracies for the SVM-based, KL, PMI and random ranking methods are reported for the top $1,000$ features. The macro-averaged accuracy when using all the features in the space without selecting relational features is 42.43%. For the verb space, the results indicate

[15] The corpus was collected by Charles Clarke at the University of Waterloo.

[16] `http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`

[17] `http://www.nltk.org`

Table 3.8: Accuracy per relation type for the top 1000 ranked features.

Relation	SVM-based	KL	Verb-space	PMI	Random
Hypernym	**73.68**	71.93	**73.68**	56.14	54.39
Meronym	**70.18**	68.42	61.4	45.61	56.14
Event	**78.95**	73.68	66.67	29.82	54.39
Attribute	**33.33**	13.33	23.33	30.00	10.00
Cause-Purpose	41.07	**44.64**	37.50	28.57	21.43
Space-Time	58.93	**64.29**	62.5	33.93	46.43
Reference	57.41	59.26	**64.81**	42.59	33.33
Macro-average	**59.08**	56.51	55.7	38.10	39.44

the performance of the $12,000$ verbs in the feature space. The classification approach for weighting the features and the verb-space perform equally for hypernym relation. For meronym, event and attribute relation types, the proposed linear-SVM outperforms other methods of feature ranking. The KL divergence-based method shows its ability to perform well compared with other methods for cause-purpose and space-time relations. Among the different relation types compared in Table 3.8, the classification-based weighting method demonstrates the highest macro-average accuracy with the other baselines. The fact that the proposed method could improve the performance for many relations of the relational classification task empirically justifies our proposal for a data-driven approach for feature selection to measure relational similarity.

We analyse the performance of the relational feature ranking methods by evaluating which of these methods ranks the relational features at the top of the list. Figure 3.8 shows the micro-average accuracies of the top-ranked features selected by the different methods; verb-space is not included in this comparison as it is not a ranking method for feature selection. We start by evaluating the top-ranked feature, subsequently adding ten more features at a time. The random baseline randomly selects a subset of features from $\mathcal{S}$. As shown in the Figure, the top-weighted features using the proposed linear SVM-based approach outperforms all other methods for the relational similarity measurement. The proposed method statistically significantly outperforms (according to the McNemar test with $p < 0.05$) all other methods for ranking the most informative features in the top-ranked feature list. This indicates that the effective features for measuring relational similarity are indeed ranked at the top by the proposed method. In addition, our results show that it is possible to maintain a relational classification accuracy while using a small subset of the features (e.g., the top 100 features). The KL divergence-based ranking method follows the classification approach for ranking the best features for relational similarity. However, the PMI-based method gives accuracies comparable with the random feature selection method. PMI is known to assign higher values to rare features, thereby preferring these features. We believe this might be an issue when selecting features for representing word-pairs.

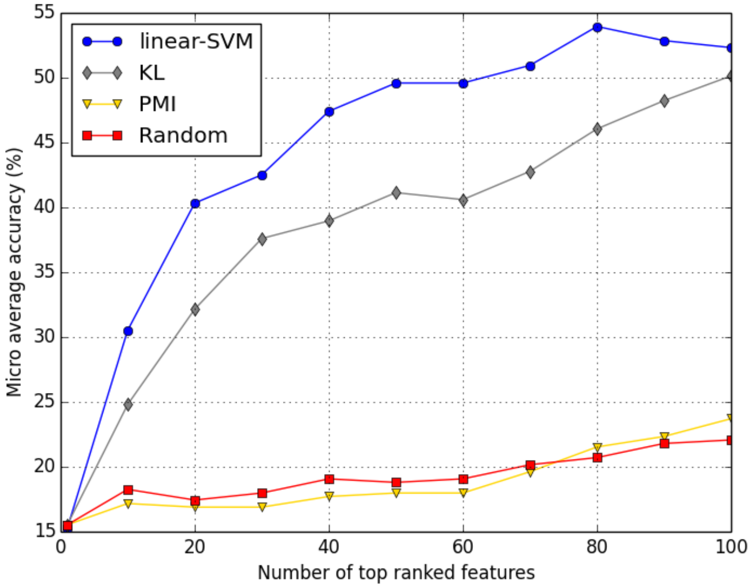


Figure 3.8: Cumulative evaluation of feature weighting methods.

3.6 Summary

This chapter has presented an exhaustive evaluation for the contribution of pre-trained word embeddings to represent relations between pairs of words. We compared unsupervised compositional operators that require no learning with the aim to derive existing relations between two words, given their word embeddings as the input. We considered four unsupervised compositional operators, namely, PairDiff, Mult, Add, and Concat. We used different pre-trained word embeddings and evaluated the performance of the operators on multiple relational tasks. We observed that PairDiff is the best linear/unsupervised operator to access relational properties across the considered tasks and the word embedding models. We also studied the effect of dimensionality on the performance of these two operators and showed that the sparsity of the input embeddings impacts the Mult operator and not the negativity of the input word embedding dimensions as speculated in prior work. The closer examination of different relation types revealed that an important portion of relational instances of lexical-semantic relations that lead to misclassification violates the primary assumption that linguistic lexical relations can be encoded under the PairDiff relational operator. This observation invites us to consider model improvements that can account for a broader range of semantic relations.

This chapter has also presented a method for discovering a discriminative feature space for measuring relational similarity from data. The relational classification results show that using labelled data to train a linear classifier for feature selection can improve the feature space in relational similarity measurements. The proposed method outperforms the KL and PMI methods for discovering relational feature spaces. Using PMI to discover relational features has been demonstrated to offer relatively poor performance, a finding that is consistent with previous work for text classification tasks (Xu et al., 2007). In addition, the classification-based weighting method reports better performance for many relation types compared with the functional verb space.

The analysis conducted in this chapter was limited to unsupervised operators in the sense that there are no parameters in the operators that can be (or must be) learnt from training data. This raises the question of whether we can learn better compositional operators from labelled data to further improve the performance of the compositional approaches for relation representations, which we explore in the coming chapters. The next chapter analyses bilinear operators between two word embeddings for the task of relation representation.

4 Mathematical Analysis of Bilinear Relation Representations

4.1 Introduction

In Chapters 2 and 3, we showed that a simple method for representing a relation between two words is to compute the difference between their corresponding word embeddings. Despite the initial success, it remains unclear as to whether PairDiff is the best operator for obtaining a relational representation from word embeddings. To this end, this chapter presents a theoretical analysis of generalised bilinear operators that can be used to measure the ℓ_2 relational distance between two word-pairs.

If we assume that the words and relations are represented by vectors embedded in some common space, then the operator we are seeking must be able to produce a vector representing the relation between two words, given their word embeddings as the only input. The space of operators that can be used to compose relational embeddings is open and vast. A space of particular interest from a computational point-of-view is the bilinear operators that can be parametrised using tensors and matrices. In this chapter, we examine operators that consider pairwise interactions between two word embeddings (second-order terms) and contributions from individual word embeddings towards their relational embedding (first-order terms). The optimality of this bilinear relational operator is evaluated using the expected ℓ_2 relational distance between analogous (positive) vs non-analogous (negative) word-pairs. The theoretical analysis provided in this chapter expands the understanding of relational embedding methods, and will inspire future research on accurate relational embedding methods using word embeddings as the input.

Bilinear models have been studied in different scenarios for relational tasks (Socher et al., 2013a; Madhyastha et al., 2014; Glavaš and Ponzetto, 2017; Glavaš and Vulić, 2018). For instance, Socher et al. (2013a) represent relations by 3-D tensors and entities as vectors in a KG, then the parameters are learnt jointly such that the triple scoring function that is generated by bilinear forms is optimised. This is different from our tasks as we aim to apply the bilinear operator on pre-trained word embeddings as a generalisation function to analyse relation representations. Madhyastha et al. (2014) also investigate probabilistic bilinear forms to perform a relation-specific prediction between two words (e.g., noun-adjective). Unlike our proposal in this chapter, they consider a different model for each relation, while we focus on the problem of a generalised relation representation model. Another variation is that they practically induce low-rank constraints on the bilinear matrix parameters to project source and target words into a lower-dimensional space in which an element-wise inner-product takes place. Likewise, for relation classification, Glavaš and Ponzetto (2017) and Glavaš and Vulić (2018) apply tensors as operators between specialised word representations to classify given word-pairs to relations, where specialised space and tensors are learnt jointly.

The organisation of this chapter is as follows. The chapter commences with Section 4.2 that provides a formal definition of bilinear operators for representing relations between words. We analyse the optimality of the bilinear operator by estimating relational distances between word-pairs under stated assumptions, as defined in Section 4.3. If we assume that word embeddings are standardised, uncorrelated and that word-pairs are independent, we prove in Section 4.4 that bilinear relational compositional operators are independent of bilinear pairwise interactions between the two input word embeddings. Moreover, under regularised settings defined in Section 4.5, the bilinear operator further simplifies to a linear combination of the input embeddings, and the expected loss over positive and negative instances becomes zero. Our theoretical analysis is supported by empirical evidence to make it tenable, as shown in Section 4.6. A discussion about the conducted analysis and a summary of the results are provided Section 4.7 and Section 4.8, respectively.

4.2 Bilinear Relational Operators

Recall that we consider the problem of representing the semantic relation r between two given words a and b. We assume that a and b are already represented in some d-dimensional space respectively by their word embeddings $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^d$. The relation between two words can be represented using different linear algebraic structures. Two popular alternatives are vectors (Nickel et al., 2016; Bordes et al., 2013; Minervini et al., 2017; Trouillon et al., 2016) and matrices (Socher et al., 2013a; Bollegala et al., 2015). Vector representations are preferred over matrix representations because of the smaller number of parameters to be learnt (Nickel et al., 2015).

Let us assume that the relation r is represented by a vector $\boldsymbol{r} \in \mathbb{R}^\delta$ in some δ-dimensional space. Therefore, we can write $\boldsymbol{r}_{ab}$ as a function f that takes two vectors (corresponding to the embeddings of the two words) as the input and returns a single vector (representing the relation between the two words) as given in (4.1).

$$f : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^\delta \tag{4.1}$$

Having both words and relations represented in the same $d = \delta$ dimensional space is useful for performing linear algebraic operations using these representations in that space. For example, in the TransE KGE model (Bordes et al., 2013), the strength of a relation r that exists between two words a and b is computed as the ℓ_1 or ℓ_2 norm of the vector $(\boldsymbol{a} + \boldsymbol{r} - \boldsymbol{b})$ using the word and relation embeddings. Such direct comparisons between word and relation embeddings would not be possible if words and relations were not embedded in the same vector space. If $\delta < d$, we can first project word embeddings to a lower δ-dimensional space using some dimensionality reduction method such as SVD, whereas if $\delta > d$ we can learn higher δ-dimensional overcomplete word representations (Faruqui et al., 2015b) from the original d-dimensional word embeddings. Therefore, we will limit our theoretical analysis in this chapter to the $\delta = d$ case for ease of description.

Different functions can be used as f that satisfy the domain and range requirements specified by (5.1). If we assume multi-linearity for relationships, f can be generally written as an operator including $\boldsymbol{a}$, $\boldsymbol{b}$ and a tensor $\underline{\mathbf{A}}$. The most general functional form of this bilinear operator is given by (4.2).

$$\boldsymbol{r}_{ab} = \boldsymbol{a}^\top \underline{\mathbf{A}} \boldsymbol{b} + \mathbf{P}\boldsymbol{a} + \mathbf{Q}\boldsymbol{b} \tag{4.2}$$

Here, $\underline{\mathbf{A}} \in \mathbb{R}^{d \times d \times \delta}$ is a 3-way tensor in which each slice is a $d \times d$ real matrix. Let us denote the k-th slice of $\underline{\mathbf{A}}$ by $\mathbf{A}^{(k)}$ and its (i, j) element by $A_{ij}^{(k)}$. The first term in (4.2) corresponds to the pairwise interactions between $\boldsymbol{a}$ and $\boldsymbol{b}$, wherein each slice in the tensor is a bilinear form that maps two input vectors to a scalar ($\mathbf{A}^{(k)} : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$). The second and the third terms in (4.2) are parametrised by $\mathbf{P}, \mathbf{Q} \in \mathbb{R}^{\delta \times d}$, which are the nonsingular projection matrices[1] involving first-order contributions of $\boldsymbol{a}$ and $\boldsymbol{b}$ towards $\boldsymbol{r}$.

4.3 Learning Settings and Assumptions

Let us consider the problem of learning the simplest bilinear functional form according to (4.2) from a given dataset of analogous word-pairs $\mathcal{D}_+ = \{((a, b), (c, d))\}_{i=1}^N$, wherein a relation in the pair (a, b) is analogous to that in (c, d). Specifically, we would like to learn the parameters $\underline{\mathbf{A}}$, $\mathbf{P}$ and $\mathbf{Q}$ such that some distance (i.e., loss) between analogous word-pairs is minimised. As a concrete example of a distance function, let us consider the

[1] If the projection matrix is nonsingular, then the inverse projection exists, which preserves the dimensionality of the embedding space.

popularly used Euclidean distance[2] (ℓ_2 loss) for two word pairs given by (4.3).

$$J((a,b),(c,d)) = ||\boldsymbol{r}_{ab} - \boldsymbol{r}_{cd}||_2^2 \tag{4.3}$$

If we were provided with only analogous word-pairs (i.e., positive examples), then this task could be trivially achieved by setting all of the parameters to zero. However, such a trivial solution would not generalise to unseen test data. Therefore, in addition to $\mathcal{D}_+$ we would require a set of non-analogous word-pairs $\mathcal{D}_-$ as negative examples. Such negative examples are often generated in prior work by randomly corrupting positive relational tuples (Bordes et al., 2013; Nickel et al., 2016; Trouillon et al., 2016) or by training an adversarial generator (Minervini et al., 2017; Cai and Wang, 2018).

The total loss J over both positive and negative training data can be written as follows:

$$J = \sum_{((a,b),(c,d)) \in \mathcal{D}_+} ||\boldsymbol{r}_{ab} - \boldsymbol{r}_{cd}||_2^2 - \sum_{((a,b),(c,d)) \in \mathcal{D}_-} ||\boldsymbol{r}_{ab} - \boldsymbol{r}_{cd}||_2^2 \tag{4.4}$$

Assuming that the training word-pairs are randomly sampled from $\mathcal{D}_+$ and $\mathcal{D}_-$ according to two distributions respectively p_+ and p_-, we can compute the total expected loss, $\mathbb{E}_p[J]$, as follows:

$$\mathbb{E}_p[J] = \mathbb{E}_{p_+}\left[||\boldsymbol{r}_{ab} - \boldsymbol{r}_{cd}||_2^2\right] - \mathbb{E}_{p_-}\left[||\boldsymbol{r}_{ab} - \boldsymbol{r}_{cd}||_2^2\right] \tag{4.5}$$

We make the following assumptions to further analyse the properties of relational embeddings considering bilinear operators.

Uncorrelation: The correlation between any two distinct dimensions i and j in the word embeddings space is defined in (4.6), and it is equal to zero (i.e., $\mathrm{corr}(w_i, w_j) = 0$).

$$\mathrm{corr}(w_i, w_j) = \frac{\mathrm{Cov}(w_i, w_j)}{\sigma_{w_i}\sigma_{w_j}} = \frac{\sum_{w \in \mathcal{V}} (w_i - \mu_{w_i})(w_j - \mu_{w_j})}{\sqrt{\sum_{w \in \mathcal{V}} (w_i - \mu_{w_i})^2}\sqrt{\sum_{w \in \mathcal{V}} \left(w_j - \mu_{w_j}\right)^2}} \tag{4.6}$$

Here, Cov is the covariance between w_i and w_j, σ_{w_i} and σ_{w_j} are respectively the standard deviations of w_i and w_j, and μ_{w_i} is the mean of w_i. Over a set of related word-pairs (a,b), cross-correlation between two distinct dimensions would be zero (i.e., $\mathrm{corr}(a_i, b_j) = 0$), whereas correlation between the same dimension of these two related words would be some positive value close to one (i.e., $\mathrm{corr}(a_i, b_i) = 1$). On the other hand, if u and v are two words that has no relation (e.g., randomly paired words), then the element-wise dimensions will have no correlation (i.e., $\mathrm{corr}(u_i, v_i) = \mathrm{corr}(u_i, v_j) = 0$). One might think that these correlations of word embedding dimensions to be

[2] For ℓ_2 normalised vectors, their Euclidean distance is a monotonously decreasing function of their cosine similarity.

strong assumptions, but we later empirically show their validity in Section 4.6.1 for a wide range of word embedding models.

Standardisation: Word embeddings are standardised to zero mean and unit variance. This is a linear transformation in the word embedding space and does not affect the relative positioning in the embedding space. In particular, translating word embeddings such that they have a zero mean has shown to improve performance in similarity tasks (Mu and Viswanath, 2018).

Relational Independence: Word pairs in the training data are assumed to be independent. For example, whether a particular semantic relation r exists between a and b, is assumed to be independent of any other relation r' that exists between c and d in a different pair. In other words, this assumption means that each of the word-pair (a, b) and (c, d) is generated independently of the other, and so their chance to coming together can be simplified to: $p\left((a,b),(c,d)\right) = p\left(a,b\right)p\left(c,d\right)$.

4.4 Theorem and Proof

Under the stated assumptions in the previous section, Theorem 1 holds for relation representations given by (4.2).

Theorem 1. *Consider the bilinear relational embedding defined by* (4.2) *computed using uncorrelated word embeddings. If the word embeddings are standardised, then the expected loss given by* (4.5) *over a relationally independent set of word pairs is independent of* $\underline{\mathbf{A}}$.

Proof. Let us consider the bilinear term in (4.2). Because i and $j (\neq i)$ dimensions of word embeddings are uncorrelated by the assumption (i.e., $\mathrm{corr}(w_i, w_j) = 0$), from the definition of correlation we have,

$$\mathrm{corr}(w_i, w_j) = \mathbb{E}[w_i w_j] - \mathbb{E}[w_i]\mathbb{E}[w_j] = 0 \tag{4.7}$$

$$\mathbb{E}[w_i w_j] = \mathbb{E}[w_i]\mathbb{E}[w_j]. \tag{4.8}$$

Moreover, from the standardisation assumption we have, $\mathbb{E}[w_i] = 0, \quad \forall i = 1 \ldots d$. From (4.8) it follows that:

$$\mathbb{E}[w_i w_j] = 0 \tag{4.9}$$

for $i \neq j$ dimensions.

We will next show that (4.5) is independent of the tensor $\underline{\mathbf{A}}$. For this purpose, let us consider the $\mathbb{E}_{p_+}$ term first and write the k-th dimension of $\boldsymbol{r}_{ab}$ using $\mathbf{A}^{(k)}$, $\mathbf{P}$ and $\mathbf{Q}$ as follows:

$$r_{ab_k} = \sum_{i,j} \left(A^{(k)}_{ij} a_i b_j \right) + \sum_n P_{kn} a_n + \sum_n Q_{kn} b_n \tag{4.10}$$

Plugging (4.10) in (4.5) and computing the expected loss over all positive training instances we get,

$$\mathbb{E}_{p+} \left[\sum_k \left(\sum_{i,j} \left(A^{(k)}_{ij} \left(a_i b_j - c_i d_j \right) \right) + \sum_n P_{kn} \left(a_n - c_n \right) + \sum_n Q_{kn} \left(b_n - d_n \right) \right)^2 \right] \tag{4.11}$$

Terms that involve only elements in $\mathbf{A}^{(k)}$ take the form:

$$\begin{aligned} &\sum_{i,j} \sum_{l,m} \mathbb{E}_{p+} \left[A^{(k)}_{ij} A^{(k)}_{lm} \left(a_i b_j - c_i d_j \right) \left(a_l b_m - c_l d_m \right) \right] \\ &= \sum_{i,j} \sum_{l,m} A^{(k)}_{ij} A^{(k)}_{lm} \left(\mathbb{E}_{p+} \left[a_i b_j a_l b_m \right] - \mathbb{E}_{p+} \left[a_i b_j c_l d_m \right] - \mathbb{E}_{p+} \left[c_i d_j a_l b_m \right] + \mathbb{E}_{p+} \left[c_i d_j c_l d_m \right] \right) \end{aligned} \tag{4.12}$$

Lets first analyse the cases where $i \neq j$ and $l \neq m$. Because of the relational independence assumption, the second and the third expectations in (4.12) can be written as follows:

$$\begin{aligned} \mathbb{E}_{p+} \left[a_i b_j c_l d_m \right] &= \mathbb{E}_{p+} \left[a_i b_j \right] \mathbb{E}_{p+} \left[c_l d_m \right] \\ \mathbb{E}_{p+} \left[c_i d_j a_l b_m \right] &= \mathbb{E}_{p+} \left[c_i d_j \right] \mathbb{E}_{p+} \left[a_l b_m \right] \end{aligned} \tag{4.13}$$

The expectations in the right hand side of (4.13) contain the product of different dimensionalities in two different words. The expected value of the product of two different dimensions in the same word is zero from (4.9). In addition, as stated in the first assumption, such cross-correlations are likely to be zero between different words. On the other hand, first and fourth expectations in (4.12) involve the same pair of words. For example, we could write the fourth expectation as follows:

$$\mathbb{E}_{p+} \left[c_i d_j c_l d_m \right] = \mathbb{E}_{p+} \left[(c_i c_l)(d_j d_m) \right] = \mathbb{E}_{p+} \left[C_{il} D_{jm} \right] \tag{4.14}$$

Here, $C_{il} = c_i c_l$ and $D_{jm} = d_j d_m$. If we think of $\boldsymbol{C}$ and $\boldsymbol{D}$ as d^2-dimensional word embeddings, $\mathbb{E}_{p+} \left[C_{il} D_{jm} \right]$ represents the expectation over two distinct dimensions of $\boldsymbol{C}$ and $\boldsymbol{D}$ for $il \neq jm$. Therefore, from the same logic as above, this expectation is approximately zero. Note that il could be equal to jm even when $i \neq j$ and $l \neq m$. However, such cases are rare minority. Nevertheless, it is an approximation and not an exact zero.

For $i = j = l = m$ case we have,

$$A_{ii}^{(k)^2} \left(\mathbb{E}_{p+}\left[a_i^2 b_i^2\right] - 2\mathbb{E}_{p+}\left[a_i b_i c_i d_i\right] + \mathbb{E}_{p+}\left[c_i^2 d_i^2\right]\right) \tag{4.15}$$

Because we are considering word-pairs (a, b) for which some relation is known to hold (i.e., we are not randomly pairing words), from the definition of the correlation between the same dimension in different words we have:

$$\begin{aligned} \text{corr}\left(a_i^2, b_i^2\right) &= \mathbb{E}\left[a_i^2 b_i^2\right] - \mathbb{E}\left[a_i^2\right]\mathbb{E}\left[b_i^2\right] = 1 \\ \mathbb{E}\left[a_i^2 b_i^2\right] &= \mathbb{E}\left[a_i^2\right]\mathbb{E}\left[b_i^2\right] + 1 \end{aligned}$$

Because $\mathbb{E}\left[a_i^2\right] = \mathbb{E}\left[b_i^2\right] = 1$ from the standardisation, we get:

$$\mathbb{E}_{p+}\left[a_i^2 b_i^2\right] = 2 \tag{4.16}$$

Lets analyse the second term in (4.15). From the relational independence and because the word embeddings are assumed to be standardised to unit variance, we obtain the follows:

$$2\mathbb{E}_{p+}\left[a_i b_i c_i d_i\right] = 2\mathbb{E}_{p+}\left[a_i b_i\right]\mathbb{E}_{p+}\left[c_i d_i\right] = 2. \tag{4.17}$$

According to (4.16) and (4.17), (4.15) evaluates to $2A_{ii}^{(k)^2}$. We will then get the same term from the negative expectations and they would cancel out as $2A_{ii}^{(k)^2}$ is independent of the training dataset.

Next, lets consider the $A_{ij}^{(k)} P_{kn}$ terms in the expansion of (4.11) given by,

$$2\sum_{i,j}\sum_{n} A_{ij}^{(k)} P_{kn}\left(a_i b_j - c_i d_j\right)\left(a_n - c_n\right). \tag{4.18}$$

Taking the expectation of (4.18) w.r.t. p_+ we get,

$$2\sum_{i,j}\sum_{n} A_{ij}^{(k)} P_{kn}\left(\mathbb{E}_{p+}\left[a_i b_j a_n\right] - \mathbb{E}_{p+}\left[a_i b_j c_n\right] - \mathbb{E}_{p+}\left[c_i d_j a_n\right] + \mathbb{E}_{p+}\left[c_i d_j c_n\right]\right). \tag{4.19}$$

For $i \neq j \neq n$ case, we can use d^2-dimensional word embeddings to write $\mathbb{E}_{p+}\left[c_i d_j c_n\right]$ as $\mathbb{E}_{p+}\left[C_{in} d_j\right]$, which is zero following the same logic as above. On the other hand, $\mathbb{E}_{p+}\left[a_i b_j c_n\right]$ is equal to $\mathbb{E}_{p+}\left[a_i b_j\right]\mathbb{E}_{p+}\left[c_n\right]$ because of the independency between (a, b) and c. These terms vanish according to the assumptions. The extreme case of $i = j = n$ leads to:

$$2\sum_{i,j}\sum_{n} A_{ij}^{(k)} P_{kn}\left(\mathbb{E}_{p+}\left[a_i^2 b_i\right] - \mathbb{E}_{p+}\left[a_i b_i c_i\right] - \mathbb{E}_{p+}\left[c_i d_i a_i\right] + \mathbb{E}_{p+}\left[c_i^2 d_i\right]\right). \tag{4.20}$$

Because $\mathbb{E}_{p+}[a_i b_i] = 1$, then we can also expect that $\mathbb{E}_{p+}[a_i^2 b_i] = 1$. Also, we can rewrite $\mathbb{E}_{p+}\left[a_i b_i c_i\right]$ as $\mathbb{E}_{p+}\left[a_i b_i\right] \mathbb{E}_{p+}\left[c_i\right]$. Thus each of the four expectations in (4.20) approach to one and would cancel out each other to be zero. It is worth noting that this case is small compared to $i \neq j \neq n$ and thus can be ignored in practice. A similar argument can be used to show that terms that involve $A_{ij}^{(k)} Q_{kn}$ disappear from (4.11).

From the provided analysis, we conclude that $\underline{\mathbf{A}}$ does not play any part in the expected loss over the training examples. Therefore, from (4.5) we see that the expected loss over the entire training dataset is independent of $\underline{\mathbf{A}}$. The next section analyses the defined loss under bilinear relational operator with regularisations. □

4.5 Analysis of the Regularised ℓ_2 loss

As a special case, if we attempt to minimise the expected loss under some regularisation on $\underline{\mathbf{A}}$ such as the Frobenius norm regularisation, then this can be achieved by sending $\underline{\mathbf{A}}$ to zero tensor because according to Theorem 1, (4.2) is independent from $\underline{\mathbf{A}}$. With $\underline{\mathbf{A}} = \underline{\mathbf{0}}$, the relation between a and b can be simplified to:

$$\boldsymbol{r}_{ab} = \mathbf{P}\boldsymbol{a} + \mathbf{Q}\boldsymbol{b} \tag{4.21}$$

Then the expected loss over the positive instances, using matrix and vector notations, is given by (4.22).

$$\begin{aligned}
&\mathbb{E}_{p+}\left[||\mathbf{P}\left(\boldsymbol{a}-\boldsymbol{c}\right) + \mathbf{Q}\left(\boldsymbol{b}-\boldsymbol{d}\right)||_2^2\right] \\
&= \mathbb{E}_{p+}\left[\left(\boldsymbol{a}-\boldsymbol{c}\right)^\top \mathbf{P}^\top \mathbf{P}\left(\boldsymbol{a}-\boldsymbol{c}\right)\right] + \mathbb{E}_{p+}\left[\left(\boldsymbol{a}-\boldsymbol{c}\right)^\top \mathbf{P}^\top \mathbf{Q}\left(\boldsymbol{b}-\boldsymbol{d}\right)\right] + \\
&\mathbb{E}_{p+}\left[\left(\boldsymbol{b}-\boldsymbol{d}\right)^\top \mathbf{Q}^\top \mathbf{P}\left(\boldsymbol{a}-\boldsymbol{c}\right)\right] + \mathbb{E}_{p+}\left[\left(\boldsymbol{b}-\boldsymbol{d}\right)^\top \mathbf{Q}^\top \mathbf{Q}\left(\boldsymbol{b}-\boldsymbol{d}\right)\right]
\end{aligned} \tag{4.22}$$

The second expectation term in the right hand side of (4.22) can be computed as follows:

$$\begin{aligned}
&\mathbb{E}_{p+}\left[\left(\boldsymbol{a}-\boldsymbol{c}\right)^\top \mathbf{P}^\top \mathbf{Q}\left(\boldsymbol{b}-\boldsymbol{d}\right)\right] \\
&= \sum_{i,j} \left(\mathbf{P}^\top \mathbf{Q}\right)_{ij} \mathbb{E}_{p+}\left[\left(a_i - c_i\right)\left(b_j - d_j\right)\right] \\
&= \sum_{i,j} \left(\mathbf{P}^\top \mathbf{Q}\right)_{ij} \left(\mathbb{E}_{p+}\left[a_i b_j\right] - \mathbb{E}_{p+}\left[a_i d_j\right] - \mathbb{E}_{p+}\left[c_i b_j\right] + \mathbb{E}_{p+}\left[c_i d_j\right]\right)
\end{aligned} \tag{4.23}$$

When $i \neq j$, each of the four expectations in the RHS of (4.26) are zero from the uncorrelation assumption between two different dimensions of related or unrelated words. When $i = j$, we have:

$$\begin{aligned}&\sum_{i,i}\left(\mathbf{P}^\top\mathbf{Q}\right)_{ii}\left(\mathbb{E}_{p_+}\left[a_ib_i\right]-\mathbb{E}_{p_+}\left[a_id_i\right]-\mathbb{E}_{p_+}\left[c_ib_i\right]+\mathbb{E}_{p_+}\left[c_id_i\right]\right)\\&=2\sum_{i,i}\left(\mathbf{P}^\top\mathbf{Q}\right)_{ii}\end{aligned}\tag{4.24}$$

The first and the forth terms will be equal to one from the correlation assumption between the same dimension of two related words. Because the same dimension in unrelated words is uncorrelated, $\mathbb{E}_{p_+}\left[a_id_i\right]$ and $\mathbb{E}_{p_+}\left[c_ib_i\right]$ evaluate to zero and we got the result in (4.24). A similar argument can be used to show that the third expectation term in the RHS of (4.22) also evaluates to $2\sum_{i,i}\left(\mathbf{Q}^\top\mathbf{P}\right)_{ii}$.

Now lets consider the first expectation term in the RHS of (4.22), which can be computed as follows:

$$\begin{aligned}&\mathbb{E}_{p_+}\left[\left(\boldsymbol{a}-\boldsymbol{c}\right)^\top\mathbf{P}^\top\mathbf{P}\left(\boldsymbol{a}-\boldsymbol{c}\right)\right]\\&=\sum_{i,j}\left(\mathbf{P}^\top\mathbf{P}\right)_{ij}\mathbb{E}_{p_+}\left[\left(a_i-c_i\right)\left(a_j-c_j\right)\right]\\&=\sum_{i,j}\left(\mathbf{P}^\top\mathbf{P}\right)_{ij}\left(\mathbb{E}_{p_+}\left[a_ia_j\right]-\mathbb{E}_{p_+}\left[a_ic_j\right]-\mathbb{E}_{p_+}\left[c_ia_j\right]+\mathbb{E}_{p_+}\left[c_ic_j\right]\right)\end{aligned}\tag{4.25}$$

When $i\neq j$, it follows from the uncorrelation assumption that each of the four expectation terms in the RHS of (4.25) will be zero. For $i=j$ case we have,

$$\begin{aligned}&\sum_{i,i}\left(\mathbf{P}^\top\mathbf{P}\right)_{ii}\left(\mathbb{E}_{p_+}\left[a_i^2\right]-2\mathbb{E}_{p_+}\left[a_ic_i\right]+\mathbb{E}_{p_+}\left[{c_i}^2\right]\right)\\&=2\sum_{i,i}\left(\mathbf{P}^\top\mathbf{P}\right)_{ii}\end{aligned}\tag{4.26}$$

Note that $\mathbb{E}_{p_+}\left[a_i^2\right]=\mathbb{E}_{p_+}\left[c_i^2\right]=1$ from the standardisation (unit variance) assumption, and $\mathbb{E}_{p_+}\left[a_ic_i\right]=0$, which gives the result in (4.26). Similarly, the fourth expectation term in the RHS of (4.22) evaluates to $2\sum_{i,j}\left(\mathbf{Q}^\top\mathbf{Q}\right)_{ii}$. Combined, (4.22) evaluates to:

$$2\sum_{i,j}\left(\left(\mathbf{P}^\top\mathbf{P}\right)_{ii}+\left(\mathbf{P}^\top\mathbf{Q}\right)_{ii}+\left(\mathbf{Q}^\top\mathbf{P}\right)_{ii}+\left(\mathbf{Q}^\top\mathbf{Q}\right)_{ii}\right)\tag{4.27}$$

Note that (4.27) is independent of the positive instances and will be equal to the expected loss over negative instances, which gives $\mathbb{E}_p[J]=0$ for the relational embedding given by (4.21).

It is interesting to note that PairDiff is a special case of (4.21), where $\mathbf{P}=\mathbf{I}$ and $\mathbf{Q}=-\mathbf{I}$. In the general case where word embeddings are nonstandardised to unit variance, we can set $\mathbf{P}$ to be the diagonal matrix where $\mathbf{P}_{ii} = 1/\sigma_i$, where σ_i is the variance of the i-th dimension of the word embedding space, to enforce standardisation. Considering that $\mathbf{P}, \mathbf{Q}$ are parameters of the relational embedding, this is analogous to *batch normalisation* (Ioffe and Szegedy, 2015), where the appropriate parameters for the normalisation are learnt during training.

4.6 Experiments and Results

To make our theory tenable, this section provides support for the proven theorem with empirical evidence. First, in Section 4.6.1, we empirically validate the uncorrelation assumption for six word embedding models. This empirical evidence implies that our theoretical analysis applies to relational representations composed from a wide range of word embedding learning methods. Then, we empirically learn the bilinear operator using analogous and non-analogous word-pairs as presented in Section 4.6.2. In Section 4.6.3, we experimentally show that a bilinear operator reaches its optimal performance in two relational analogy benchmark datasets when it satisfies the requirements of the PairDiff operator.

4.6.1 Cross-dimensional Correlations

A key assumption in our theoretical analysis is the uncorrelations between different dimensions in word embeddings. Here, we empirically verify the uncorrelation assumption for different input word embeddings. We make use of SG, CBOW, GloVe and LSA(SVD) word embeddings that we trained on ukWaC corpus as described in Section 3.2. In addition, we examine two other popular models to obtain semantic representations for words, namely, Latent Dirichlet Allocation and Hierarchical Sparse Coding, as described in the following two sections.

Latent Dirichlet Allocation

In the context of NLP, Latent Dirichlet Allocation (LDA) is an unsupervised probabilistic topical model that typically aims to extract hidden themes (topics) in a collection of documents. LDA has been developed by Blei et al. (2003), and since then it has seen many areas of application such as document classification and sentiment analysis. The main idea is to model the similarity of documents in terms of what these documents are about (topic). Topical models such as LDA seek to imitate a human ability in classifying documents into topics based on the words appearing in these documents. Briefly, LDA represents each document as a probability distribution over a latent (small number) of topics, and each topic is characterised by a distribution over the words in the vocabulary. Document-topic and topic-word distributions are derived in such a way that they best explain the observed unlabelled textual documents. Conceptually, the LSA model can be seen as a topic model as it shares the underlying assumption that observations consist of a mixture themes, such that a space of reduced dimensionality through linear algebra represents latent themes.

The LDA model starts with a pre-processing steps on a large collection of documents to obtain a document-word matrix in which each element corresponds to a term-frequency vs inverse document frequency between a document and a word. The words that make the document-word matrix are the only observable features for the LDA model to be trained to produce a topic distribution that describes how a document could be generated. The number of latent topics and how these topics are assigned to a document are hyper-parameters to be defined. The two important parameters for the Dirichlet distribution are α and β that control per-document topic distribution and per-topic word distribution, respectively. A low value of α, and similarly for β, means that a document probably belongs to a few of the topics. The generative process for a corpus goes as follows. First, the model chooses the probabilities over the K topics for the M documents $\boldsymbol{\theta}_{d=1,\dots,M}$ using Dirichlet distribution parametrised by α. Similarly for topics, where the mixing proportions over words $\boldsymbol{\varphi}_{k=1,\dots,K}$ are drawn from Dirichlet(β). Then, for the j^{th} position in the i^{th} document, the model choose a topic z_{ij} from a multinomial distribution with $\boldsymbol{\theta}_i$ parameters. Finally, a word w_{ij} is assigned to the selected topic based on the probability vector over the words for z_{ij} (i.e., $\boldsymbol{\varphi}_{z_{ij}}$). Combined, LDA is learnt to find the topic distributions and the words associated to each topic, which are likely to generate the collection of documents.

As explained above, LDA was originally proposed to represent similarity between documents. However, we are interested in getting word representations for our task. A number of studies adopted the LDA model to generate semantic spaces for words (Mitchell and Lapata, 2010; Liu et al., 2015). Thus, we built an LDA model to represent each word by its distribution over the set of topics. Ideally, each topic will capture some semantic category and the topic distribution provides a semantic representation for a word. For our experiments, we prefer to use English articles in Wikipedia as a corpus to train our LDA because it is highly contextualised since each article normally covers a single topic. We use gensim[3] to extract latent topics from a 2017 January dump of English Wikipedia. The hyper-parameters are set as follows: number of topics=50, $\alpha = 0.02$, β=0.1, vocabulary size=$100,000$.

Hierarchical Sparse Coding

In contrast to the above-mentioned word embeddings, which are dense and flat structured, we evaluate Hierarchical Sparse Coding (HSC) that is used to produce sparse and hierarchical word embeddings (Yogatama et al., 2015). Inspired by considering the hierarchically-

[3] Gensim is an open source topic modelling framework made in Python: `https://radimrehurek.com/gensim/wiki.html`

organised lexicons such as WordNet, the authors propose an embedding model for words considering hierarchical structure between word embedding dimensions, which can be seen as organising the latent concepts in the embedding space. Specifically, given the high-dimensional context-word matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ capturing PMI between the occurrences of m contexts and n words, $\mathbf{X}$ is factorised under sparse coding such that the reconstruction loss given in (4.28) is minimised.

$$\arg\min_{\mathbf{D},\mathbf{A}} ||\mathbf{X} - \mathbf{DA}||_2^2 + \lambda \Omega\left(\mathbf{A}\right) \tag{4.28}$$

Here, $\mathbf{D} \in \mathbb{R}^{m \times d}$ and $\mathbf{A} \in \mathbb{R}^{d \times n}$ are the dictionary of basis vectors and the code matrix, respectively (where d indicates the number of latent dimensions). Ω is the structured regulariser for $\mathbf{A}$, that is applied to each column (a word) in $\mathbf{A}$.

In our experiments, we employ the publicly available 52-dimensional HSC word embeddings that are generated from a forest of four tree structures among the 52 latent dimensions[4].

Correlation Results

Given a word embedding matrix $\mathbf{W} \in \mathbb{R}^{n \times d}$, where each row correspond to the d-dimensional embedding of a word in a vocabulary containing n words, we compute a correlation matrix $\mathbf{C} \in \mathbb{R}^{d \times d}$, where the (i, j) element, C_{ij}, denotes the Pearson correlation coefficient between the i-th and j-th dimensions in the word embeddings over the n words in the vocabulary. By construction $C_{ii} = 1$ and the histograms of the cross-dimensional correlations $(i \neq j)$ are shown in Figure 4.1 for 50 dimensional word embeddings obtained from the six methods described above. The mean of the absolute pairwise correlations for each embedding type and the standard deviation (sd) are indicated in the figure. From Figure 4.1, irrespective of the word embedding learning method used, we see that cross-dimensional correlations are distributed in a narrow range with an almost zero mean.

In addition to the correlations between dimensions considering the same set of words, we validate the cross-correlations and element-wise correlations between two sets of related words as required in the conducted analysis. To do so, we collect the word pairs from all the benchmarks datasets that have been used in this thesis, which were introduced in Section 3.3. In total, we have a set $\mathcal{P}$ of $18,791$ related word-pairs (a, b). Then, we generate two embedding matrices $\mathbf{S}, \mathbf{T} \in \mathbb{R}^{|\mathcal{P}| \times d}$ such that $\mathbf{S}$ includes embeddings for the words that act as the source arguments of the word-pairs (i.e., a), whereas $\mathbf{T}$ is constructed for the corresponding target words (i.e., b). We then compute a correlation matrix $\mathbf{C} \in \mathbb{R}^{d \times d}$, where C_{ij} here denotes the Pearson correlation coefficient between the i-th dimension (column) in $\mathbf{S}$ and j-th dimension (column) in $\mathbf{T}$ over the word-pairs in $\mathcal{P}$. Figure 4.2 shows the heatmap

[4] `http://www.cs.cmu.edu/~ark/dyogatam/wordvecs/`

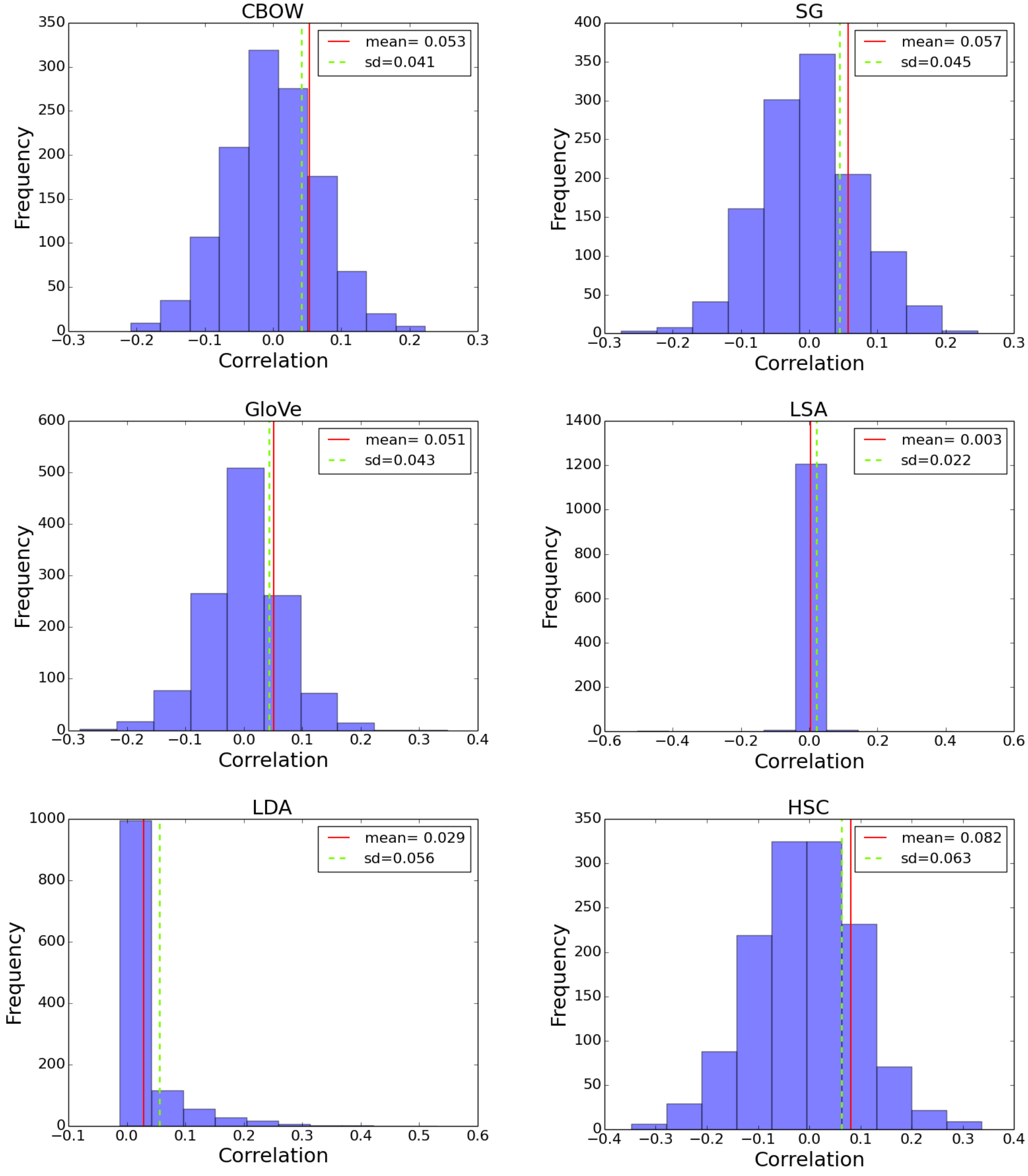
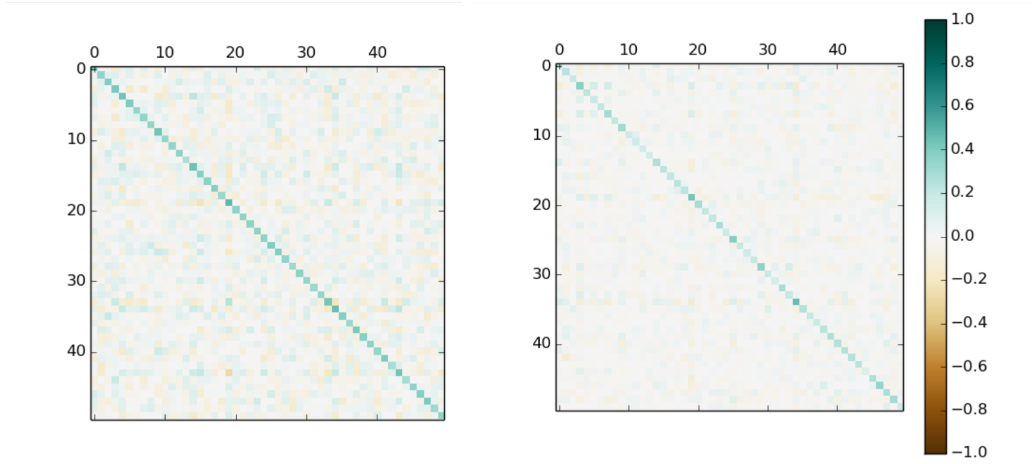


Figure 4.1: Cross-dimensional correlations for six word embedding models.

for the Pearson correlation scores between the dimensions of **S** and **T** using 50-dimensional SG word embeddings. As shown in the figure, element-wise correlations (i.e., the diagonal values of **C**) are some positive correlations compared to cross-correlations that are close to zero. The means and standard deviations of the absolute element-wise (i.e., $\text{corr}(a_i, b_i)$) and cross-correlations (i.e., $\text{corr}(a_i, b_j)$) for all the embedding models are presented in Table 4.1. All the embedding models report means of zero for cross-correlations with small standard



(a) $\mathrm{corr}(a_i, b_j)$

(b) $\mathrm{corr}(a_i^2, b_j^2)$

Figure 4.2: Heatmap of the Pearson correlation scores between (a) SG embedding dimensions and (b) SG embedding squared dimensions among related word pairs (a, b).

Table 4.1: Means and standard deviations (mean±sd) for Pearson correlation scores between dimensions in word embeddings for related word-pairs.

Correlation types	CBOW	SG	GloVe	LSA	LDA	HSC
$\mathrm{corr}(a_i, b_j), i \neq j$	0.056±0.043	0.053±0.041	0.051±0.038	0.039±0.032	0.009±0.016	0.056±0.045
$\mathrm{corr}(a_i, b_i)$	0.371± 0.053	0.369±0.044	0.331±0.042	0.343±0.084	0.261±0.226	0.324±0.060
$\mathrm{corr}(a_i^2, b_j^2), i \neq j$	0.021±0.013	0.025±0.022	0.024±0.020	0.022±0.019	0.011±0.012	0.024±0.026
$\mathrm{corr}(a_i^2, b_i^2)$	0.213 ±0.043	0.244±0.071	0.200±0.064	0.197±0.056	0.190±0.13	0.187±0.052

deviations, whereas the element-wise correlations are relatively high up to 0.371 in CBOW embeddings. On the other hand, element-wise and cross-correlations for randomly paired words shows zero means across all the word embedding models (0.005 ± 0.003). We also evaluate randomly pairing a with c, a with d, b with c and b with d, wherein (a, b) and (c, d) are instances of the same relation. We use SemEval dataset for this experiment, which reports means of 0.079 ± 0.035 and 0.022 ± 0.021 for element-wise and cross-correlation in SG[5], respectively. These small means and sd supports the uncorrelation assumption of unrelated pairs.

These results empirically validate the uncorrelation assumption we used in our theoretical analysis. Moreover, this result indicates that Theorem 1 can be applied to a wide-range of existing word embedding models. In the next section, the bilinear relation representation that minimises the defined loss in (4.4) is learnt using real-world training data.

[5] All other embedding models show roughly similar correlation scores for this experiment.

4.6.2 Learning Relation Representations

Our theoretical analysis in Section 4.4 shows that the performance of the bilinear relational embedding is independent of the tensor operator $\underline{\mathbf{A}}$. To empirically verify this claim, we conduct the following experiment. For this purpose, we use the BATS dataset that was introduced in Section 3.3.5 that contains 40 semantic and syntactic relation types, and generate positive examples by pairing word-pairs that have the same relation types. Approximately each relation type has 1,225 word-pairs, which enables us to generate a total of 48,000 positive training instances (analogous word-pairs) of the form $((a, b), (c, d))$. For each pair (a, b) related by a relation r, we randomly select pairs (c, d) with a different relation type r', according to the ℓ_2 distance between the two pairs to create negative (non-analogous) instances. We generate ten negative instances from each word-pair in our experiments. We collectively refer both positive and negative training instances as the *training* dataset. In total, we collect $49,000$ analogous word-pairs and about $20,000$ non-analogous pairs.

Using the $d = 50$ dimensional word embeddings from CBOW, SG, GloVe, LSA, LDA, and HSC models, we learn relational embeddings according to (4.2) by minimising the ℓ_2 loss defined in (4.4). To avoid overfitting, we perform ℓ_2 regularisation on $\underline{\mathbf{A}}$, $\mathbf{P}$ and $\mathbf{Q}$ are regularised to diagonal matrices $p\mathbf{I}$ and $q\mathbf{I}$, for $p, q \in \mathbb{R}$. We initialise all parameters by uniformly sampling from $[-1, +1]$ and use Stochastic Gradient Descent (SGD) with AdaGrad (Duchi et al., 2011) with initial learning rate set to 0.01.

Figure 4.3 shows the Frobenius norm of the tensor $\underline{\mathbf{A}}$ (on the left vertical axis) and the values of p and q (on the right vertical axis) for the six word embeddings. In all cases, we see that as the training progresses, $\underline{\mathbf{A}}$ goes to zero as predicted by Theorem 1 under regularisation. Moreover, we see that approximately $p \approx -q = c$ is reached for some $c \in \mathbb{R}$ in all cases, which implies that $\mathbf{P} \approx -\mathbf{Q} = c\mathbf{I}$, which is the PairDiff operator. Among the six input word embeddings compared in Figure 4.1, HSC has the highest mean correlation (0.082), which implies that its dimensions are correlated more than in the other word embeddings. This is to be expected by design because a hierarchical structure is imposed on the dimensions of the word embedding during training. However, HSC embeddings also satisfy the $\underline{\mathbf{A}} \approx \underline{\mathbf{0}}$ and $p \approx -q = c$ requirements, as expected by the PairDiff. This result shows that the claim of Theorem 1 is empirically true even when the uncorrelation assumption is mildly violated.

In the next section, we test the generalisation of the learnt parameters of the bilinear operator on relational benchmark datasets.

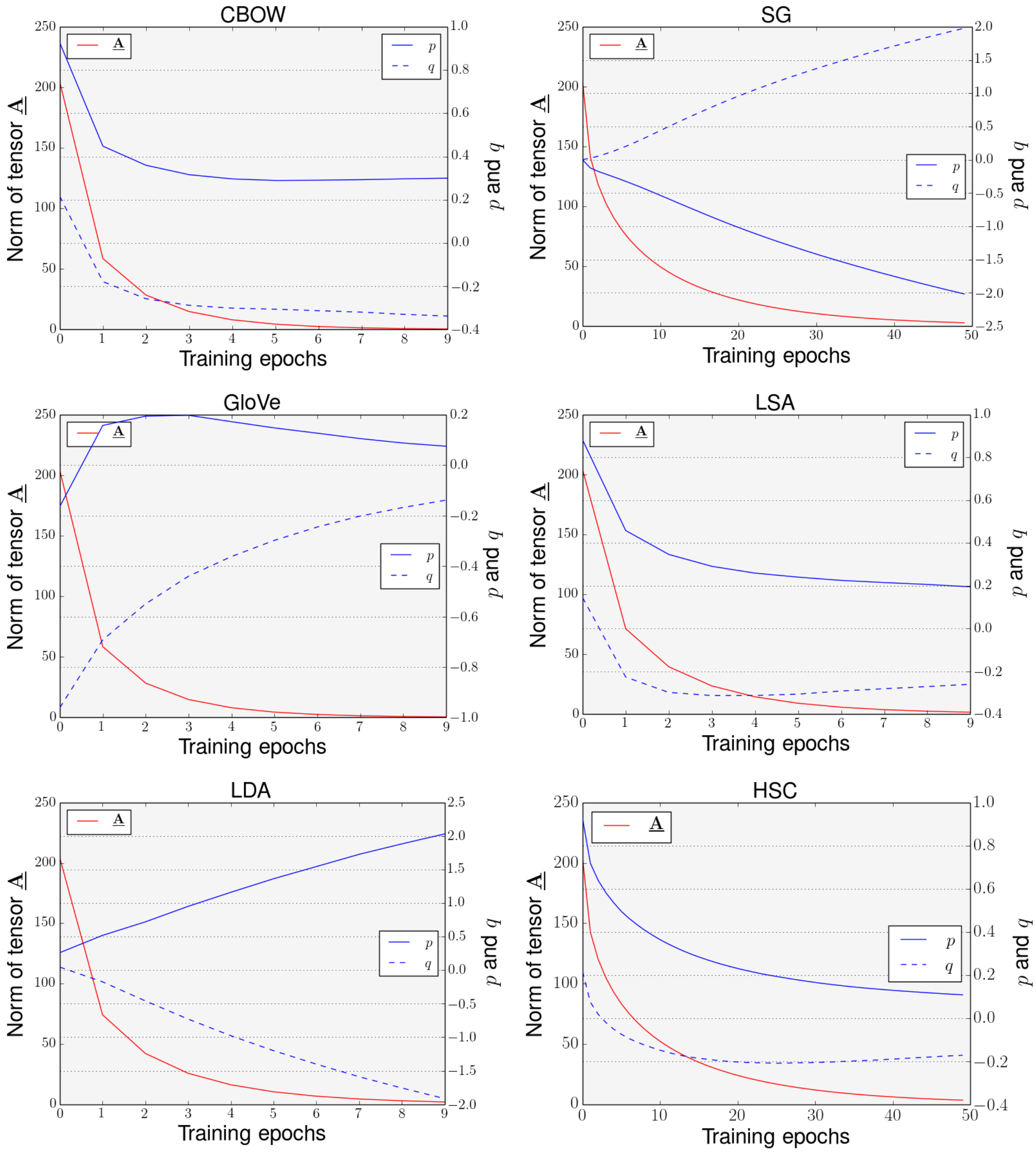


Figure 4.3: The learnt model parameters for different word embeddings of 50 dimensions.

4.6.3 Generalisation of Performance on Analogical Tasks

So far we have seen that the bilinear relational representation given by (4.2) does indeed converge to the form predicted by our theoretical analysis for different types of word embeddings. However, it remains unclear whether the parameters learnt from the training instances generated from the BATS dataset accurately generalise to other benchmark datasets for analogy detection. To emphasize, our focus here is not to outperform relational

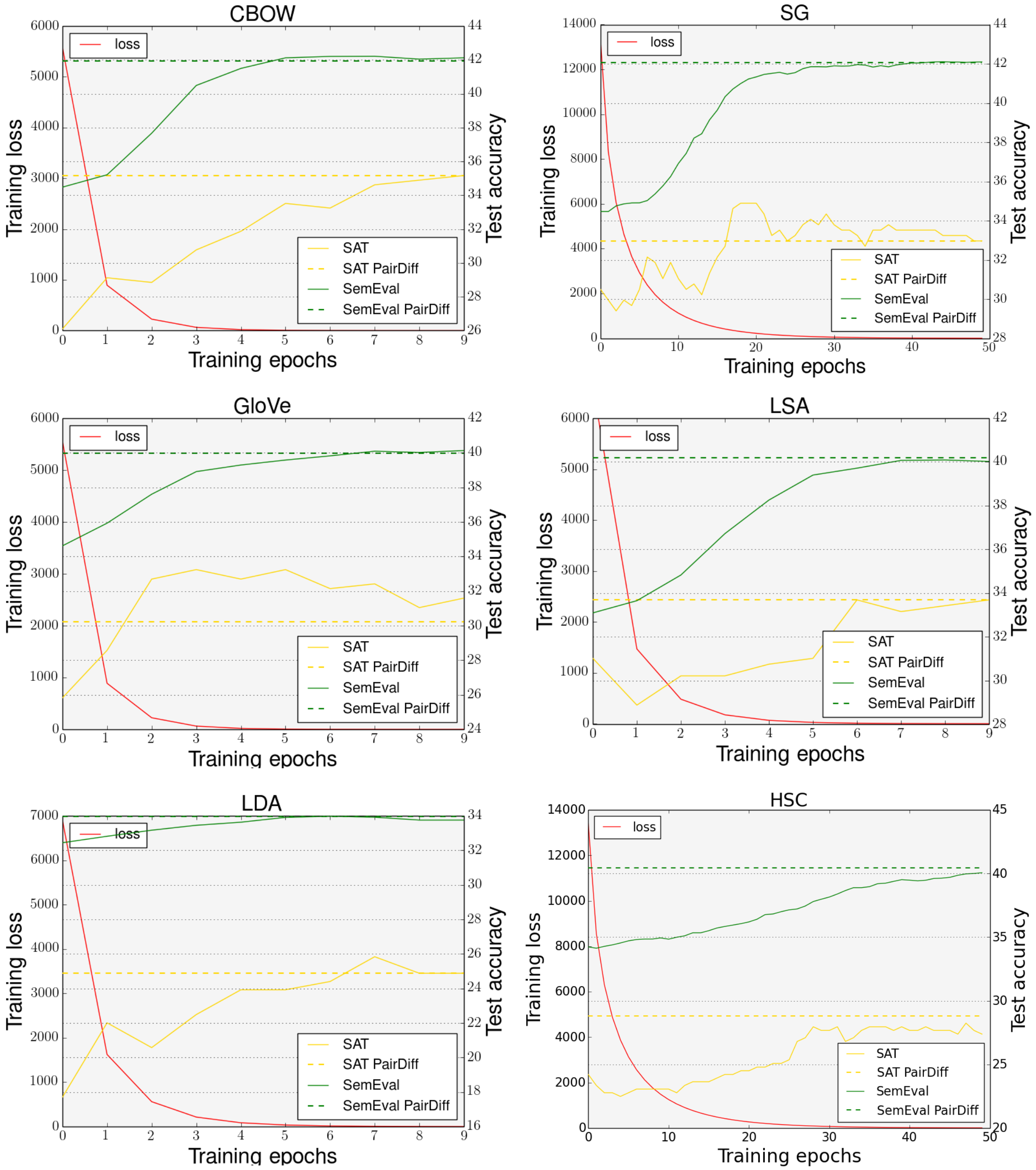


Figure 4.4: The training loss and test performance on the SAT and SemEval benchmark datasets for relational embeddings.

representation methods proposed in previous works, but rather to empirically show that the learnt operator converges to the performance of the popular PairDiff operator for the analogy detection task. To measure the generalisation capability of the learnt relational embeddings from BATS, we evaluate their performance on two other benchmark datasets: the SAT (Section 3.3.1) and the SemEval 2012-Task2 with MaxDiff metric (Section 3.3.2).

Note that we *do not* retrain $\underline{\mathbf{A}}$, $\mathbf{P}$ and $\mathbf{Q}$ in (4.2) on SAT nor SemEval, but simply use the values learnt from BATS because the purpose here was to evaluate the generalisation of the learnt operator.

Figure 4.4 shows the performance of the relational embeddings composed from 50-dimensional word embeddings across different models. Similar trends were observed for all six word embedding types. In CBOW, for example, the level of performance reported by the PairDiff operator on the SAT and SemEval datasets are respectively 35.16% and 41.94%, and are shown by horizontal dashed lines. From Figure 4.4, we see that the training loss decreases gradually with the number of training epochs and the performance of the relational embeddings on SAT and SemEval datasets reach that of the PairDiff operator. This result indicates that the relational embeddings learnt not only converge to PairDiff operator on training data but also generalise to unseen relation types in SAT and SemEval datasets.

4.7 Discussion

It is worth noting that the theoretical analysis provided in this chapter has some limitations. First, the conducted analysis does not hold when the cross-dimensional correlations in the word embeddings are not small. Although we were unable to find a word embedding learning method that violates this uncorrelation assumption in our experiments, the set of word embedding learning methods is an open and a continuously growing one. Further theoretical studies are required to consider the cases where the cross-correlations between different dimensions in a word embedding can not be ignored.

Another flaw is that the proof used the assumption that word-pairs are independent. Even though we validate the proven theorem in our experiments, in practice this assumption is questionable. For example, semantic relations are not always independent. One possible solution to avoid such an assumption is to analyse the ℓ_1 absolute loss rather than ℓ_2 least square loss.

In this chapter, we model relations as vectors and we measure the relational strength using Euclidean distance. We are aware that there are many other relation representation methods and relational strength measurement methods besides what we have considered in the paper. Similar analysis can be conducted in follow-up work for different types of relation representations and strength measures. For instance, an interesting future research direction of this work is to extend the theoretical analysis to nonlinear relation composition operators, such as for nonlinear neural networks.

4.8 Summary

This chapter presented a theoretical analysis of the bilinear operator for representing relations between words using their embeddings. We showed that, if the word embeddings are standardised and under dimensional correlation assumptions, then the expected ℓ_2 distance between analogous and non-analogous word-pairs is independent of bilinear terms, and the relation embedding further simplifies to the popular PairDiff operator under regularised settings. Among diverse methods for calculating word embeddings, we empirically verified the validity of the correlation assumptions in word embedding dimensions, which is one of the prerequisites for simplifying the bilinear operator to a linear one. Empirically, we supported the theoretical analysis by showing that when optimising a general bilinear formulation on a labeled word pair relational dataset, the solution converges to the simple linear form, and more specifically to the simple PairDiff formulation.

The next chapter will introduce proposed methods of learning compositional operators for relation representations by employing neural networks on unsupervised word embeddings. The motivation is to exploit word embeddings that capture global contexts of words and learnt using a large unlabelled text corpus to develop a supervised (or self-supervised) method from a small set of labelled data to represent relations between words.

5 Learning Compositional Operators for Relation Representations

5.1 Introduction

Despite the initial hype of the PairDiff method for relations, multiple independent works have raised concerns on word embeddings capturing relational structural properties (Linzen, 2016; Schluter, 2018; Liu et al., 2017; Rogers et al., 2017; Gladkova et al., 2016). Although PairDiff performs well on the Google analogy dataset, its performance on other relation types has been poor (Chen et al., 2017; Vylomova et al., 2016; Köper et al., 2015). Vylomova et al. (2016) tested for the generalisation ability of PairDiff using different relation types and found that semantic relations are captured less accurately compared to syntactic relations. Likewise, Köper et al. (2015) showed that word embeddings are unable to detect paradigmatic relations such as Hypernym, Synonym and Antonyms. Another reported problem of PairDiff is the bias towards attributional similarities between individual words rather than relational similarities as it fails in the presence of nearest neighbours (Rogers et al., 2017). Various issues of the PairDiff relational representation method are already covered in detail in Section 2.5.4.

Considering the above-mentioned limitations of unsupervised relation representation methods, a natural question that arises is whether it is possible to learn *data-driven* relation representation methods to overcome those limitations. This chapter addresses this research question in different ways. Briefly, we model the task of relation representation as learning a parametrised function such that we can accurately represent the relation between two given words from their word representations. We refer to these functions as compositional operators for relation representations. The underlying idea is that word embeddings are learnt in an unsupervised manner using a large corpus tend to include features that are correlating with semantic relations between words. As such, we can apply supervised approaches on a small set of data to create relation representations between words.

The chapter is organised as follows. Section 5.2 presents a supervised relation representation method that is based on word embeddings for related word-pairs and their relation labels. In short, we train a multi-class relation prediction neural network and adopt the penultimate layer as relation representations of the given word-pairs. Experimental results show that our proposed model can generalise by representing relations of word-pairs from unseen relation types, outperforming different baselines such as PairDiff. Then, in Section 5.3, compositional methods that learn relational operators are regularised during training with relational patterns of word-pairs. We call such methods Context-Guided self-supervised Relation Embeddings. Empirical findings of ranking word-pairs by measuring relational similarity confirm that the proposed context-guided model improves relation representations. A summary of the work considered in this chapter and some conclusions are presented in Section 5.4.

5.2 Learning Supervised Relation Compositional Operators

We model relation representation as learning a parametrised operator $f(a, b; \theta)$ such that we can accurately represent the relation between two given words a and b from their word representations $\boldsymbol{a}$ and $\boldsymbol{b}$, without modifying the input word embeddings[1]. For this purpose, we propose a Multi-class Neural Network Penultimate Layer (MnnPL), a simple and effective parametrised operator for computing relation representations from word representations. Specifically, we train a nonlinear multilayer feed-forward neural network using a labelled dataset consisting of word-pairs for different relation types, where the task is to predict the relation between two input words represented by their pre-trained word embeddings. We find that the penultimate layer of the trained neural network provides an accurate relation representation that generalises beyond the relations in the training dataset. It is worth noting that our focus here is not to classify a given pair to a relation in a pre-defined set (relation classification), but rather to obtain a good representation for the relation between the two words in the pair.

The aforementioned strategy of modelling relations is similar to that in Rossiello et al. (2019)'s study. In particular, the authors evaluate their proposed analogy detection neural network on novel relation types as in our work. They also evaluated the effectiveness of the penultimate layer of their analogy model to provide word-pair representations. Unlike our model, their method is considered to be pattern-based relation representation since it requires sentences in which two words co-occur for both training and testing instances. However, our proposed MnnPL is compositional as we do not need of co-occurrence sentences.

[1] The word embeddings are not updated because at evaluation time, we will generate relation representation between words that might never be seen during training and thus their embeddings never get tuned.

This section is organised as follows. The proposed MnnPL is introduced in Section 5.2.1. We evaluate the relation embeddings learnt by the proposed MnnPL on two standard tasks: out-of-domain relation prediction and measuring the degree of relational similarities between two word-pairs. In Section 5.2.2, we stated the experimental setup that we follow to train the proposed method. In Section 5.2.3 and 5.2.4, we discuss the experiments conducted on the out-of-domain and in-domain relation prediction task, respectively. In Section 5.2.5, we evaluate relation embeddings by measuring correlation with relational similarity judgments.

5.2.1 Multiclass Neural Network Penultimate Layer

Our goal is to learn a parametrised two-argument operator $f(\cdot, \cdot; \theta)$ that can accurately represent the relation between two given words a and b using their pre-trained d-dimensional word embeddings $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^d$. Here, θ denotes the set of parameters that governs the behaviour of f, which can be seen as a *supervised* operator that outputs a relation representation from two input word representations. The output of f, for example, could be a vector that exists in the same or a different vector space as $\boldsymbol{a}$ and $\boldsymbol{b}$, as given by (5.1).

$$f(\boldsymbol{a}, \boldsymbol{b}; \theta) : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^\delta \tag{5.1}$$

In general $d \neq \delta$, and word and relation representations can have different dimensionalities; even when $d = \delta$ they might be in different vector spaces. We could extend this definition to include higher-order relation representations such as matrices or tensors, but doing so would increase the computational overhead. Therefore, we limit supervised relational operators such that they return vectors as given by (5.1). We note that unsupervised relational operators such as PairDiff and vector concatenation are specific instances of this definition. For example, for PairDiff we have $f(\boldsymbol{a}, \boldsymbol{b}; \theta) = \boldsymbol{a} - \boldsymbol{b}$ $(d = \delta)$, and for vector concatenation we have $f(\boldsymbol{a}, \boldsymbol{b}; \theta) = \boldsymbol{a} \oplus \boldsymbol{b}$ $(\delta = 2d)$, where $\oplus$ denotes the concatenation of two vectors. In unsupervised operators, θ is a constant that does not influence the output relation embedding.

Having been provided with a dataset $\mathcal{D} = \{(a, b, r)_1, \ldots, (a, b, r)_N\}$ containing N word-pairs (a, b) labelled with relation types $r \in \{1, \ldots, |\mathcal{R}|\}$ from a set of relations $\mathcal{R}$, we train a neural network to predict r given the concatenated pre-trained word embeddings $\boldsymbol{a} \oplus \boldsymbol{b}$ as the input. We implement the proposed supervised relation composition operator, MnnPL, as a feed-forward neural network with two hidden layers followed by a softmax layer as shown in Figure 5.1. Mathematically, the input $(\boldsymbol{i})$, the hidden $(\boldsymbol{h})$ and the output $(\boldsymbol{o})$ layers along with the loss function $\mathcal{L}(\mathcal{D}, \theta)$ are defined as follows:

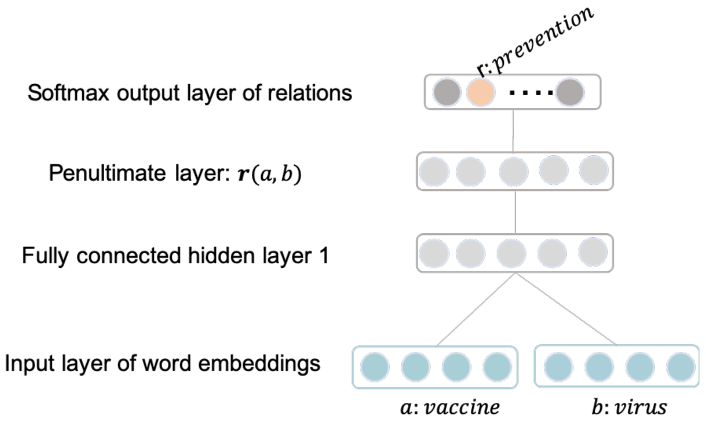

r: prevention
Softmax output layer of relations
Penultimate layer: $r(a,b)$
Fully connected hidden layer 1
Input layer of word embeddings
a: vaccine
b: virus

Figure 5.1: Architecture of the proposed MnnPL, a feed forward neural network that is used to model the supervised relational operator f.

$$\begin{aligned}
&\boldsymbol{i} = \boldsymbol{a} \oplus \boldsymbol{b}\\
&\boldsymbol{h}_1 = g\left(\mathbf{W}_1 \boldsymbol{i} + \boldsymbol{s}_1\right)\\
&\boldsymbol{h}_2 = g\left(\mathbf{W}_2 \boldsymbol{h}_1 + \boldsymbol{s}_2\right)\\
&\boldsymbol{o} = \mathbf{W}_3 \boldsymbol{h}_2 + \boldsymbol{s}_3\\
&\hat{\boldsymbol{p}} = \mathrm{softmax}(\boldsymbol{o}) = \frac{\exp(o_i)}{\sum_{j=1}^{|\mathcal{R}|} \exp(o_j)} \quad \text{for} \quad i = 1, \ldots, |\mathcal{R}|\\
&\mathcal{L}(\mathcal{D}, \theta) = -\frac{1}{N} \sum_{(a,b,r) \in \mathcal{D}} \log\left(\hat{p}_r\right) + \frac{\lambda}{2} ||\theta||_2^2
\end{aligned} \tag{5.2}$$

Weight matrices for the hidden layers are $\mathbf{W}_1$ and $\mathbf{W}_2$, whereas the bias vectors are $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$. g refers to the nonlinear activation function for the hidden layers. We experiment with different nonlinearities in the hidden layers. The output layer is the softmax over the relation labels of a given dataset, which is parametrised by $\mathbf{W}_3$ and $\boldsymbol{s}_3$. We minimise the ℓ_2 regularised softmax cross-entropy loss over the training instances as defined in (5.2). Here, θ is the set of the learnable parameters in MnnPL, $\theta = \mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, \boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{s}_3$. As the model's name implies, MnnPL, after training a single model parametrised by the neural network for a set of relations, we use the penultimate layer (i.e., the output of the final hidden layer $\boldsymbol{h}_2$) as the relation representation for a word-pair.

We emphasise that our goal is *not* to classify a given pair into a specific set of relations, but rather to find a representation of the relation between any pair of words. Therefore, we test the learnt relation representation operator using relations that are not seen during training (i.e., out-of-domain examples) by holding out a subset of relations during training. Combined, our method can be seen as an instance of transfer learning, which typically aims to use the knowledge gained when learning one task in a source domain (relation classification) and transferring it to a different, but related, task (relation representation) on a target domain. Our evaluation can be also considered as a zero-shot learning setting (Larochelle et al., 2008), where some classes are not available during training the model and they are only given at inference time. The section below describes the training settings for the proposed model in detail.

5.2.2 Training Setup

Datasets. We used two previously proposed datasets for evaluating the proposed MnnPL: BATS (Section 3.3.5) and DiffVec (Section 3.3.4). In the DiffVec dataset, we exclude ATTRIBUTE:Action-ObjectAttribute relation from the experiments as it has less than ten instances. We experimented with 50-dimensional CBOW, SG, GloVe and LSA(SVD) word embedding models trained on the ukWaC as the input to the neural network (introduced in Section 3.2). Overall, we found ℓ_2 normalisation of word embeddings to improve results.

Implementation details. The size of the input layer of the MnnPL is $2d$, where d is the dimensionality of the input word embeddings. We set the size of each hidden layer in MnnPL to d ($= 50$), and thus the dimensionality of the relation embeddings δ from the penultimate layer is equal to d. We use SGD with Momentum (Qian, 1999) with a mini-batch size of 128 to minimise the ℓ_2 regularised cross-entropy error. All parameters are initialised by uniformly sampling from $[-1, +1]$ and the initial learning rate is set to 0.1. Dropout regularisation is applied with a 0.25 rate. TensorFlow[2] was used to implement the model. All hyperparameters are tuned using a randomly selected 10% of training data, set aside as a validation dataset. Specifically, we selected the number of the hidden layers among {1, 2, 3}, the activation function g of the hidden layers among {tanh, relu, linear}, and the ℓ_2 regularisation coefficient from {0.1, 0.01, 0.001} via grid search within the validation dataset. We found the optimal configuration was to set the number of hidden layers to two and the nonlinear activation to tanh (hyperbolic tangent function). The tanh function squashes the input between -1 and 1, and is defined as follows: $\tanh(x) = \frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)}$. The optimal ℓ_2 regularisation coefficient λ was 0.001. These settings performed consistently well in all our evaluations.

Comparison methods. As our main baselines, we use the unsupervised operators for representing relations of word-pairs that are studied in Section 3.4.1, namely : PairDiff, Concatenation (Concat), elementwise addition (Add) and elementwise multiplication (Mult). These operators are referred to as unsupervised in the sense that there are no parameters in those operators that can be learnt from the training data. We also compare the proposed

[2] TensorFlow is an open source platform for machine learning: `https://www.tensorflow.org`

MnnPL with the bilinear operator proposed in Chapter 4 as a supervised relation representation method. For convenience, the bilinear operator is restated in (5.3). We refer to this operator as BiLin.

$$\boldsymbol{r}_{ab} = \boldsymbol{a}^{\top}\underline{\mathbf{A}}\boldsymbol{b} + \mathbf{P}\boldsymbol{a} + \mathbf{Q}\boldsymbol{b} + \boldsymbol{s} \tag{5.3}$$

Here, $\underline{\mathbf{A}} \in \mathbb{R}^{d \times d \times \delta}$ is a 3-way tensor in which each slice is a $d \times d$ real matrix. $\mathbf{P}, \mathbf{Q} \in \mathbb{R}^{\delta \times d}$ are the projection matrices of $\boldsymbol{a}$ and $\boldsymbol{b}$, respectively. We train the BiLin operator using a margin-based rank loss objective. Specifically, we minimise the distance between the relation representations of the analogous pairs (positive instances), while maximising the distance between the representations of non-analogous examples (negative instances) created via random perturbations. Given a set of word pairs $\mathcal{S}_r$ that are related by the same relation, we generate positive training instances $((a,b),(c,d))$ by pairing word-pairs $(a,b) \in \mathcal{S}_r$ and $(c,d) \in \mathcal{S}_r$. To generate negative training instances, we corrupt a positive instance by pairing $(a,b) \in \mathcal{S}_r$ with a word-pair $(c',d') \in \mathcal{S}_{r'}$ that belongs to a different relation $r' \neq r$. One negative instance is generated for each analogous example in our experiments, resulting in a balanced binary labelled dataset $\mathcal{D}$. The regularised training objective for BiLin $\mathcal{L}(\mathcal{D};\theta)$ is given by (5.4).

$$\sum_{((a,b),(c,d),(c',d'))\in\mathcal{D}} \max(0, \mu + ||\boldsymbol{r}_{ab} - \boldsymbol{r}_{cd}||_2^2 - ||r_{ab} - \boldsymbol{r}_{c'd'}||_2^2) + \frac{\lambda}{2}\,||\theta||_2^2 \tag{5.4}$$

Here, μ is a margin hyperparameter set to 1 according to the best accuracy on the validation dataset. The best regularisation coefficient λ for the tensor $\underline{\mathbf{A}}$ on the validation dataset was 0.1. However, regularising $\mathbf{P}$ and $\mathbf{Q}$ decreased the performance on the validation set, and therefore were not regularised. Considering 50-dimensional word embeddings, the dimensionality of BiLin parameters are set to: $\underline{\mathbf{A}} \in \mathbb{R}^{50\times50\times50}$, $\mathbf{P} \in \mathbb{R}^{50\times50}$ and $\mathbf{Q} \in \mathbb{R}^{50\times50}$.

The supervised methods are trained for a maximum of 1000 epochs, wherein the best model is selected by early stopping through evaluating the performance on the validation set. When the performance ceases to improve for 15 consecutive epochs, we stop the training and use the last best saved model. The following section discusses the out-of-domain relation representation task.

5.2.3 Evaluating Out-of-Domain Relations

A critical evaluation criterion for a relation representation learning method is whether it can accurately represent not only the relations that exist in the training data that was used to learn the relation representation but can also generalise to unseen relations (*out-of-domain)*. Therefore, to evaluate the different relation representation methods, we employ them in an out-of-domain relation prediction task. Specifically, we use different relations for testing than that used in training. No training is required for unsupervised operators.

Evaluation Protocol

Here, we describe the evaluation protocol of out-of-domain task in detail. Lets denote a set of relation types by $\mathcal{R}$ and a set of word-pairs covering the relations in $\mathcal{R}$ by $\mathcal{D}$. First, we randomly sample five target relations from the dataset to construct a relation set $\mathcal{R}_t$ for testing and the remainder represents a set of source relations $\mathcal{R}_s$ that is used for training the supervised relational operators including the BiLin and the proposed MnnPL. We use the set $\mathcal{D}_s$ of word-pair instances covering $\mathcal{R}_s$ to learn the supervised operators by predicting the relations in $\mathcal{R}_s$. To evaluate the performance of such operators, we use the relational instances in the test split $\mathcal{D}_t$ that cover the out-of-domain relations in $\mathcal{R}_t$. We conduct 1-NN relation classification on the $\mathcal{D}_t$ dataset. The task is to predict the relation that exists between two words a and b from the sampled relations in $\mathcal{R}_t$. Specifically, we represent the relation between two words using each relational operator on the corresponding word embeddings. Next, we measure the cosine similarity between representations for the stem pair and all the word-pairs in $\mathcal{D}_t$. For each target word-pair, if the top-ranked word-pair has the same relation as the stem pair, then it is considered to be a correct match. Note that we do *not* use $\mathcal{D}_t$ for learning or updating the (supervised) relational operators but use it only for the 1-NN relation predictor. We repeat this process ten times by selecting different $\mathcal{R}_s$ and $\mathcal{R}_t$ (four or five relations as targets) relation sets and use leave-one-out evaluation for the 1-NN as the evaluation criteria. We compute the (micro-averaged) classification accuracy of the test sets as the evaluation measure. Because each relation type in an out-of-domain relation set has multiple relational instances, a suitable relation representation method retrieves the related pairs for a target pair at the top of the ranked list. For this purpose, we measure Mean Average Precision (MAP) for the relation representation methods. MAP is the mean of the Average Precision (AP) for each test word-pair, which is computed considering a ranked list of candidate word-pairs as follows:

$$\mathrm{AP} = \frac{\sum_{k=1}^{K} \left(\frac{\text{\# matches pairs in top } k}{k} \right) \times \mathcal{I}(k)}{\text{\#correct pairs}} \tag{5.5}$$

where $\mathcal{I}(k)$ is an indicator function that returns 1 if a relation label of a candidate pair at rank k matches the label of the given test pair, 0 otherwise.

To derive further insights into the relation representations learnt, following Nastase et al. (2013), we use the notion of "near" vs. "far" analogies considering the similarities between the corresponding words in the two related pairs. For example, (*tiger*, *feline*), (*cat*, *animal*) and (*motorcycle*, *vehicle*) are all instances of the is-a-hypernym relation. One can see that (*tiger*, *feline*) is closer to (*cat*, *animal*) than (*motorcycle*, *vehicle*). Here, *tiger* and *cat* are similar

Table 5.1: The two nearest and the two farthest word-pairs for some stem word-pairs together with their similarity scores according to (5.6).

Relation type	Stem pair	Nearest to Farthest
Hypernym	(food:cherry)	(fruit:plum)$_{0.87}$,(veggie:parsley)$_{0.81}$, . . . ,(artifact:helicopter)$_{0.43}$
Space-Time	(theatre:play)	(hall:music)$_{0.69}$,(studio:art)$_{0.68}$, . . . ,(mine:coal)$_{0.38}$
Cause-Effect	(disease:sickness)	(illness:discomfort)$_{0.84}$,(headache:stress)$_{0.78}$, . . . ,(digging:hole)$_{0.47}$
Contiguity	(wall:shelf)	(sill:window)$_{0.78}$,(railing:stair)$_{0.76}$, . . . ,(margin:paper)$_{0.59}$

because they are both animals; also *feline* and *animal* have shared attributes. On the other hand, the corresponding words in the two pairs (*tiger*, *feline*) and (*motorcycle*, *vehicle*) have low attributional similarities between *tiger* and *motorcycle* or between *feline* and *vehicle*. Detecting near analogies using word embeddings is easier compared to far analogies because attributional similarity can be measured accurately using word embeddings. For this reason, we evaluate the accuracy of a relation representation method at different degrees of the analogy as follows. Given two word-pairs, we compute the cross-pair attributional similarity using SimScore defined by (5.6).

$$\text{SimScore}((a,b),(c,d)) = \frac{1}{2}(\text{sim}(\boldsymbol{a},\boldsymbol{c}) + \text{sim}(\boldsymbol{b},\boldsymbol{d})) \tag{5.6}$$

Here, $\text{sim}(\boldsymbol{x},\boldsymbol{y})$ is the cosine similarity between $\boldsymbol{x}$ and $\boldsymbol{y}$. Next, we sort the word-pairs in descending order of their SimScores (i.e., from near to far analogies). Examples of far and near analogies with SimScores for some selected word-pairs are presented in Table 5.1. To alleviate the effect of attributional similarity between two word-pairs in our evaluation, we remove the 25% top-ranked (nearest) pairs for each stem pair. Consequently, a relation representation method that relying only on attributional similarity is unlikely to accurately represent the relations between words.

Experimental Results

The average accuracy (Acc) and the MAP of the relation representation operators for CBOW, SG, GloVe and LSA embeddings on DiffVec and BATS datasets are presented in Table 5.2. As can be observed among the different embedding types, MnnPL consistently outperforms all other methods with respect to both Acc and MAP score. The differences between MnnPL and other methods for all rounds and target relations are statistically significant ($p < 0.01$) according to a paired t-test. CBOW embeddings report the best Acc and MAP scores for the two datasets in contrast to all other embedding models. The reported performance of the proposed MnnPL over the best unsupervised operator (i.e., PairDiff) has a significant improvement on DiffVec compared to BATS. One of the reasons

Table 5.2: Average accuracy and MAP of 1-NN relation classification for different relation representation methods on DiffVec and BATS datasets. Results are shown for CBOW, SG, GloVe and LSA word embeddings (50 dimensional embeddings).

	CBOW				SG			
	DiffVec		BATS		DiffVec		BATS	
Method	Acc	MAP	Acc	MAP	Acc	MAP	Acc	MAP
PairDiff	0.398	0.344	0.688	0.525	0.349	0.305	0.607	0.454
Concat	0.173	0.347	0.325	0.518	0.147	0.316	0.250	0.446
Add	0.164	0.302	0.321	0.479	0.159	0.288	0.269	0.412
Mult	0.179	0.213	0.330	0.286	0.206	0.24	0.289	0.287
BiLin	0.395	0.357	0.710	0.587	0.332	0.309	0.604	0.485
MnnPL	**0.486**	**0.421**	**0.721**	**0.624**	**0.411**	**0.373**	**0.625**	**0.522**
	GloVe				LSA			
	DiffVec		BATS		DiffVec		BATS	
Method	Acc	MAP	Acc	MAP	Acc	MAP	Acc	MAP
PairDiff	0.365	0.312	0.663	0.516	0.295	0.306	0.624	0.510
Concat	0.139	0.300	0.361	0.520	0.122	0.300	0.298	0.482
Add	0.161	0.276	0.347	0.462	0.132	0.266	0.312	0.442
Mult	0.199	0.225	0.323	0.278	0.179	0.198	0.385	0.335
BiLin	0.355	0.325	0.668	0.557	0.268	0.294	0.622	0.543
MnnPL	**0.456**	**0.381**	**0.698**	**0.585**	**0.360**	**0.342**	**0.658**	**0.59**

could be due to a small number of training examples in BATS versus DiffVec. In light with the proven theorem in Chapter 4 about bilinear operators for relation representations, we can observe that the obtained results support the theoretical analysis as for most of the cases BiLin does not significantly outperform PairDiff.

To further evaluate the accuracy of the relational operators on different relation types, we break down the evaluation per major relation type in the BATS dataset as shown in Table 5.3. For semantic relation types, we can see that lexicographic relation representations perform weaker than encyclopaedic relations for all the considered methods. More importantly, the proposed MnnPL consistently, and often substantially, outperforms the other methods for both types of semantic relations. In particular, the performance of PairDiff on lexicographic relations is poor, whereas MnnPl reports the best results. On the other hand, we found that our proposed MnnPL performs lower than unsupervised PairDiff when it comes to morphological relations.

Table 5.3: Break down of the performance for the four major relation types in the BATS dataset per method using GloVe embeddings.

	Encyclopedic		Lexicographic		Inflectional		Derivational	
Method	Acc	MAP	Acc	MAP	Acc	MAP	Acc	MAP
PairDiff	0.764	0.613	0.297	0.213	**0.905**	**0.703**	**0.789**	**0.615**
Concat	0.724	0.792	0.146	0.302	0.307	0.539	0.244	0.454
Add	0.774	0.781	0.199	0.311	0.153	0.384	0.180	0.337
Mult	0.464	0.294	0.170	0.202	0.382	0.412	0.301	0.260
BiLin	0.813	0.694	0.366	0.300	0.763	0.628	0.694	0.543
MnnPL	**0.884**	**0.813**	**0.414**	**0.338**	0.817	0.686	0.759	**0.615**

Further Analysis

Because most NLP models employ the pre-trained Glove and CBOW 300-dimensional vectors that are publicly available, it is worth showing the performance of these pre-trained models with the proposed MnnPL. We consider GloVe that is trained on Common Crawl dataset[3] (42 billion tokens), and CBOW trained on Google News[4] (100 billion words). Table 5.4 illustrates the performance of relation representation methods using the pre-trained GloVe and CBOW. In addition to the fine-grained DiffVec of 36 relations (DiffVec-fine), we examine 15 coarse-grained relations (DiffVec-coarse) in which the out-of-domain relation learning task is more challenging due to the lack of correlations between $\mathcal{R}_s$ and $\mathcal{R}_t$. Consistently, pre-trained GloVe and CBOW behave similarly to the embeddings we trained on ukWaC corpus and used extensively throughout the thesis. More interestingly, MnnPL shows comparable results for the held-out relations in DiffVec-coarse setting, which confirms that the proposed method can be transferred to resource-lean target relations without any training instances.

To compare the performance of the relation representation methods across different out-of-domain target relation sets, Table 5.5 presents the 1NN accuracy for all the five rounds from the DiffVec and BATS. For the DiffVec dataset, which includes lexicographic semantic relations, the MnnPL consistently outperforms the baselines across most of the relations individually and reports the best average for each randomly selected target set for GloVe and CBOW embeddings There are some relations where all the methods fail to represent using only word embeddings, such as Sign: Significant in which all the methods perform poorly and the difference of PairDiff with the MnnPL is small 0.031. The best accuracy reported for this relation is 0.152 with MnnPL operator on GloVe embeddings.

In the case of BATS dataset, there are differences as the relational instances in this set

[3] https://nlp.stanford.edu/projects/glove/

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

Table 5.4: Results for the out-of-domain evaluation using pre-trained 300-dimensional GloVe and CBOW.

	Pre-trained GloVe						Pre-trained CBOW					
	DiffVec-coarse		DiffVec-fine		BATS		DiffVec-coarse		DiffVec-fine		BATS	
Method	Acc	MAP	Acc	MAP	Acc	MAP	Acc	MAP	Acc	MAP	Acc	MAP
PairDiff	0.300	0.280	0.311	0.317	0.622	0.516	0.230	0.282	0.275	0.321	0.520	0.412
Concat	0.029	0.204	0.107	0.263	0.386	0.559	0.063	0.264	0.128	0.315	0.423	0.570
Add	0.057	0.185	0.113	0.238	0.387	0.505	0.086	0.226	0.123	0.273	0.444	0.538
Mult	0.151	0.206	0.171	0.241	0.344	0.306	0.150	0.214	0.125	0.195	0.494	0.412
BiLin	0.282	0.274	0.376	0.327	0.633	0.517	0.343	0.317	0.358	0.319	0.601	0.524
MnnPL	**0.415**	**0.380**	**0.470**	**0.410**	**0.706**	**0.569**	**0.522**	**0.457**	**0.520**	**0.425**	**0.723**	**0.622**

are classified to various relation types and a method that works for one relation type might not be suitable for others. To be consistent in the evaluation, we excluded two Encyclopedic (E) relations and two Lexicographics (L) for each target set $\mathcal{R}_t$. Overall, as in Table 5.5, MnnPL achieves the best results for the most of L relations and for the average 1NN accuracy for the five different target sets in the two word embeddings types. There are few exceptions as can be seen in Table 5.5. In CBOW embeddings, for $\mathcal{R}_t^{(3)}$ set Mult shows the best average 0.579 compared to 0.532 of the MnnPL. We can observe that E relations are easier than L relations, as all the methods can gain higher performance on E than L relations. Despite that, MnnPL reports good performance for both types of semantic relations.

Effect of Lexical Overlaps

As elaborated in Section 2.5.4, PairDiff is biased towards the attributional similarity between words when two word-pairs are compared. To evaluate the effect of this, we group test cases in the DiffVec dataset into two categories: (a) **lexical-overlap** (i.e., there are test cases that have one word in common between two word-pairs) and (b) **lexical-nonoverlap** (i.e., no words are common between the two word-pairs in all the test cases). In other words, given the test word-pair (a, b), then if there is a train word-pair (a, c), (b, c), (c, a) or (c, b) we consider this case in the lexical-overlap set. For example, (*animal*, *cat*) and (*animal*, *dog*) has lexical-overlap because *animal* is a common word in the two pairs. Figure 5.2 shows the average 1-NN classification accuracy for the best unsupervised operator PairDiff and MnnPL. We see that the performance drops significantly from lexical-overlap to lexical-nonoveralp by ca. 10% for PairDiff, whereas that drop is ca. 1.8% for MnnPL. This result indicates that MnnPL is affected less by attributional similarity compared to PairDiff.

Table 5.5: 1NN classification accuracy for relations in different target relation sets in out-of-domain evaluation using pre-trained GloVe and CBOW. Best performance for each relation type in each embedding model is shown in bold.

DiffVec		Pre-trained Glove						Pre-trained CBOW					
Round	Target relations	PairDiff	Concat	Add	Mult	BiLin	MnnPL	PairDiff	Concat	Add	Mult	BiLin	MnnPL
$\mathcal{R}_t^{(1)}$	Location-Process:Product	.370	.000	.037	.074	.481	**.593**	.481	.074	.037	.185	.630	**.704**
	EnablingAgent: Object	.206	.000	.000	.000	.441	**.500**	.294	.235	.324	.088	.353	**.618**
	Light-Verb-Construction	.897	.983	.914	.810	**1.0**	.914	.810	.914	.586	.397	.414	**.948**
	Sign: Significant	.242	.000	.000	.091	.273	**.455**	.091	.030	.061	.121	.121	**.424**
	Cause:CompensatoryAction	**.500**	.107	.143	.071	.286	.357	.393	.286	.214	.214	.393	**.643**
	Average	.443	.218	.219	.209	.496	**.564**	.414	.308	.244	.201	.382	**.667**
$\mathcal{R}_t^{(2)}$	Object: TypicalAction(n.v)	.152	.000	.000	.182	.152	**.455**	.333	.061	.000	.030	.483	**.485**
	Sign: Significant	.121	.000	.030	.030	.121	**.152**	.061	.000	.061	.091	.091	**.121**
	Knowledge	**.926**	.593	.333	.333	.667	.667	**.926**	.704	.370	.296	.704	.741
	Time-Action:Activity	.206	.059	.147	.294	**.441**	.235	.265	.176	.176	.147	**.294**	**.294**
	Instrument:Goal	.069	.000	.000	.034	.000	**.310**	.034	.000	.000	.034	**.241**	.207
	Average	.295	.130	.102	.175	.276	**.364**	.324	.188	.121	.120	.363	**.370**
$\mathcal{R}_t^{(3)}$	Object: TypicalAction(n.v)	.152	.000	.000	.333	.182	**.515**	.242	.000	.000	.000	**.485**	.424
	Object:State (n.n)	.250	.031	.031	.062	.219	**.562**	.031	.000	.062	.062	.125	**.562**
	Hypernym	.420	.000	.020	.040	.700	**.700**	.260	.000	.040	.200	.400	**.680**
	Attachement	.185	.259	.259	.296	.037	**.407**	.074	.074	.185	.074	.111	**.407**
	Instrument:Goal	.069	.000	.000	.069	.172	**.483**	.034	.000	.000	.000	.241	**.276**
	Average	.215	.058	.062	.160	.262	**.533**	.128	.015	.057	.067	.272	**.47**
$\mathcal{R}_t^{(4)}$	Event	.540	.060	.120	.200	.580	**.780**	.440	.060	.100	.040	.740	**.820**
	Location-Process:Product	.259	.000	.037	.037	.519	**.704**	.407	.000	.000	.111	.444	**.593**
	EnablingAgent: Object	.059	.000	.000	.118	.088	**.265**	.000	.029	.088	.059	.353	**.412**
	Instrument:Goal	.172	.000	.000	.103	.345	**.621**	.103	.034	.069	.069	.343	**.552**
	Plan	.171	.057	.143	.143	**.286**	.200	.171	.086	.286	.229	.229	**.543**
	Average	.240	.023	.060	.120	.364	**.514**	.224	.042	.109	.102	.422	**.584**
$\mathcal{R}_t^{(5)}$	Location:AssociatItem	.031	.000	.000	.062	**.250**	.094	.094	.000	.000	.188	**.281**	.125
	LocationAction:Activity	**.667**	.048	.000	.095	**.667**	**.667**	**.762**	.095	.048	.048	.571	.619
	Meronym	.180	.000	.000	.040	.120	**.300**	.047	.000	.000	.040	.160	**.500**
	Hypernym	.460	.020	.020	.080	.560	**.60**	.280	.000	.040	.100	.400	**.500**
	Representation	.067	.000	.133	**.300**	.233	**.300**	.133	.000	.100	.167	.300	**.400**
	Average	.281	.014	.031	.115	.366	**.392**	.262	.019	.038	.109	.342	**.429**
BATS													
$\mathcal{R}_t^{(1)}$	country-capital(E01)	**1.0**	.560	.000	.580	.860	.720	**.757**	.405	.135	.432	.514	.622
	country-language(E02)	**1.0**	.720	.180	.200	.980	.920	**.861**	.278	.167	.222	.472	.472
	meronyms-member(L05)	.520	.240	.340	.460	**.580**	**.580**	.531	.163	.286	.367	.571	**.878**
	hypernyms-misc(L02)	.500	.200	.240	.200	.540	**.800**	.420	.360	.460	.380	.580	**.820**
	Average	**.755**	.430	.190	.360	.740	**.755**	.642	.302	.262	.350	.534	**.698**
$\mathcal{R}_t^{(2)}$	country-capital(E01)	**1.0**	.980	.880	.640	.960	**1.0**	.784	**.973**	**.973**	**.973**	**.973**	**.973**
	animal-sound(E07)	**.740**	.060	.040	.080	.680	.580	**.880**	.500	.320	.340	.640	.760
	hypernyms-animals(L01)	.500	.140	.080	.120	.380	**.580**	.444	.6	.378	.889	.844	**.933**
	hyponyms-misc(L03)	.400	.220	.200	.220	.600	**.740**	.260	.220	.300	.600	.440	**.860**
	Average	.660	.350	.300	.265	.655	**.725**	.592	.573	.493	.701	.724	**.882**
$\mathcal{R}_t^{(3)}$	UK_city-county(E03)	.800	.600	.740	.580	.740	**.960**	.560	.560	**.640**	.480	.320	.360
	country-capital(E01)	**1.0**	.880	.820	.420	.820	.860	.595	.892	.919	**.946**	**.946**	.703
	antonyms-binary(L10)	.320	.480	**.560**	**.560**	**.560**	.440	.200	.340	.420	.380	.400	**.760**
	synonyms-exact(L08)	.380	.000	.000	.160	.320	**.400**	.184	.000	.020	**.510**	.143	.306
	Average	.625	.490	.530	.430	.610	**.665**	.385	.448	.500	**.579**	.452	.532
$\mathcal{R}_t^{(4)}$	UK_city-county(E03)	.920	.960	**1.0**	.640	.920	.840	.560	**1.0**	**1.0**	.800	.880	.880
	animal-young(E06)	.340	.020	.460	.040	.280	**.680**	.440	.380	.500	.440	.640	**.740**
	meronyms-part(L06)	.200	.000	.000	.100	.120	**.240**	.152	.000	.022	.174	.196	**.326**
	synonyms-exact(L08)	.220	.020	.040	**.560**	.260	.360	.163	.041	.041	**.469**	.347	.429
	Average	.420	.250	.375	.335	.395	**.530**	.329	.355	.391	.471	.516	**.594**
$\mathcal{R}_t^{(5)}$	animal-shelter(E08)	.640	.180	.560	.080	.740	**.900**	.920	.740	.780	.440	.800	**.94**
	name-nationality(E04)	**1.0**	.900	.820	.220	**1.0**	**1.0**	.875	.958	**1.0**	.917	.958	**1.0**
	antonyms-gradable(L09)	.600	.520	.580	.720	**.860**	.640	.640	.540	.740	.400	.920	**1.0**
	meronyms-substance(L04)	.360	.040	.200	.300	.460	**.800**	.592	.327	.490	.245	.796	**.918**
	Average	.650	.410	.540	.330	.765	**.835**	.757	.641	.753	.501	.869	**.965**

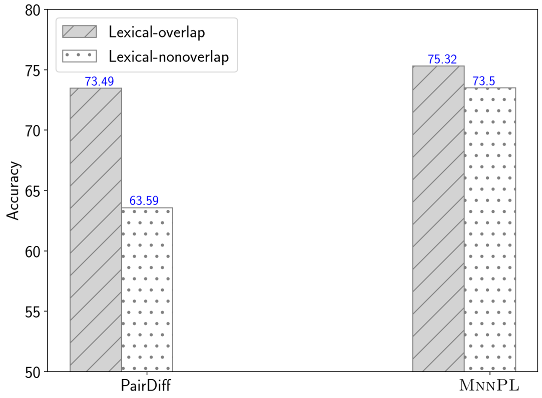


Figure 5.2: Effect of lexical overlaps in measuring word-pairs relational similarity.

5.2.4 Evaluating In-Domain Relations

We evaluate the performance of the relation representation operators considering the in-domain setting, wherein we test the performance on relational instances that belong to relation types used in the training set. Recall that $\mathcal{R}$ and $\mathcal{D}$ refer to the set of relations and the set of relational instances covering such relations, respectively. In the in-domain setting, we do not need to split $\mathcal{R}$ to source and target relation sets. Instead, we implement 5-stratified folds cross-validation considering the set of relational instances in the dataset $\mathcal{D}$. We used 1-NN and MAP metrics for the evaluation. So the in-domain experiment setting is very similar to the out-of-domain experiment except in the latter we use $\mathcal{R}_s \neq \mathcal{R}_t$ for the evaluation. Detailed results for in-domain evaluation are presented in Table 5.6. As shown in the table, MnnPL reports the best results for the in-domain setting for the two datasets. As expected, the performance for the in-domain setting is significantly better than the out-of-domain setting.

5.2.5 Measuring the Degree of Relational Similarity

Recall that the relational similarity is the correspondence between the relations that exists in two word-pairs. To measure a relational similarity score between two pairs of words, one must first identify the relation in each pair to perform such a comparison. Suitable relation embeddings should correlate highly with human judgments of relational similarity between word-pairs. For this task, we use the dataset proposed by Chen et al. (2017)[5] which is inspired by SemEval-2012 task 2 dataset (Jurgens et al., 2012). In this dataset, humans are asked to score pairs of words directly focusing on a comparison between instances with similar relations. For examples, in Location:Item relation, the pairs (*cupboard*, *dishes*) and (*kitchen*, *food*) are assigned a higher relational similarity score (6.18) than the pairs

[5] `https://github.com/sdawnchen/vector-space-analogy-analysis`

Table 5.6: 1-NN relation classification results for in-domain setting.

	CBOW				SG			
	DiffVec		BATS		DiffVec		BATS	
Method	Acc	MAP	Acc	MAP	Acc	MAP	Acc	MAP
PairDiff	0.686	0.386	0.484	0.329	0.621	0.334	0.399	0.263
Concat	0.717	0.385	0.417	0.284	0.673	0.336	0.344	0.240
Add	0.573	0.323	0.303	0.227	0.524	0.296	0.261	0.196
Mult	0.480	0.268	0.182	0.119	0.453	0.275	0.182	0.123
BiLin	0.700	0.520	0.599	0.492	0.648	0.464	0.462	0.349
MnnPL	**0.797**	**0.656**	**0.600**	**0.483**	**0.765**	**0.619**	**0.497**	**0.379**

	GloVe				LSA			
	DiffVec		BATS		DiffVec		BATS	
Method	Acc	MAP	Acc	MAP	Acc	MAP	Acc	MAP
PairDiff	0.662	0.371	0.442	0.288	0.642	0.339	0.398	0.279
Concat	0.672	0.349	0.385	0.261	0.667	0.345	0.344	0.260
Add	0.516	0.299	0.282	0.205	0.534	0.301	0.270	0.213
Mult	0.423	0.261	0.177	0.108	0.460	0.256	0.226	0.151
BiLin	0.692	0.510	0.501	0.383	0.694	0.511	0.438	0.366
MnnPL	**0.796**	**0.655**	**0.531**	**0.416**	**0.785**	**0.639**	**0.479**	**0.387**

(*cupboard*, *dishes*) and (*water*, *ocean*), which is rated 3.8. Instances of this relation can be expressed by multiple patterns such as "X *holds* Y" or "Y *in the* X", and one reason that the second example is assigned low score is that the words in the pair (*water*, *ocean*) are ordered reversely compared to other pairs. Chen et al. (2017) dataset consists of 6,194 word-pairs across 20 semantic relations. We calculated the relational similarity score of two pairs as the cosine similarity between the corresponding relation vectors generated by the considered operators. Then, we measured the Pearson correlation coefficient between the average human relational similarity ratings and the predicted scores by the methods. For this task, we choose to train the supervised methods on BATS as the overlap of the relation set between BATS and Chen datasets are small. We exclude any word-pairs in the Chen dataset that appears in the training data.

Table 5.7 shows Pearson correlations for all the four embedding models and the relational representation methods across all relations, where high values indicate a better agreement with the human notion of relational similarity. As can be observed, the proposed MnnPL correlated better with human ratings than the supervised and unsupervised baselines. According to the Fisher transformation test of statistical significance (Fisher, 1915), the reported correlations of MnnPL is statistically significant at the 0.05 significance level.

Table 5.7: Results of measuring relational similarity scores (Pearson's correlations).

Method	MnnPL	BiLin	PairDiff	Concat	Add	Mult
CBOW	**0.309**	0.258	0.172	0.277	0.223	0.204
GloVe	**0.263**	0.207	0.161	0.208	0.147	0.021
SG	**0.251**	0.176	0.161	0.208	0.147	0.021
LSA	**0.266**	0.199	0.154	0.245	0.197	0.190

Interestingly, the Concat baseline shows a stronger correlation coefficient than PairDiff. Moreover, for CBOW and LSA embeddings, Add and Mult are considered stronger than PairDiff. Consistent with the out-of-domain relation prediction task, CBOW embeddings perform better than other embeddings for measuring the degree of relational similarity. Indeed, measuring the degree of relational similarity is a challenging task and required qualified fine-grained relation embeddings to obtain accurate scores of relational instances.

5.3 Generalising Co-occurrences Between Word-Pairs and Patterns to Unseen Pairs

Although the supervised operator MnnPL that is proposed in the previous section has shown to be effective in representing relations of word-pairs, we argue that the information contained in co-occurring patterns can still provide guidance to such a supervised operator to enhance relation representation quality. As already described in Chapter 2, two main approaches can be identified in the literature for representing semantic relations between two words: *pattern-based* and *compositional* approaches. The pattern-based approaches use lexical patterns in which two words of interest co-occur, while the compositional approaches, on the other hand, attempt to represent the relation between two words from their word embeddings. Each of these approaches has drawbacks. While the main problem of the pattern-based approach is data sparseness, methods that rely only on word embeddings for the related pairs suffer from lack of relational information. Prior work on relation embeddings has predominantly focused on either one type of those two resources exclusively.

We believe that word embeddings and co-occurrence contexts collectively provide complementary information for the purpose of learning relation embeddings. For example, Bollegala et al. (2010) observed a *duality* between word-pair and pattern-based approaches for representing relations where they refer to the former as an *intentional* definition of relation representation and the latter an *extensional* definition of relation representation. The authors used this duality to propose a sequential co-clustering algorithm for discovering relations from a corpus. In prior work on relation representation learning, however, the two types of information sources are often used independently. Thus, the question of whether we can learn more expressive and superior relation representations by combining the two sources of information needed to be explored.

Hybrid approaches for relations aim to balance between the data sparsity in the pattern-based approach and the lack of relational information in the compositional methods. However, few recent studies have been devoted to incorporate the two types of information to improve the relation representations (refer to Section 2.6). In response to this gap, this section presents a proposed relation representation method that uses the contextual information from a text corpus to generalise the learnt operator for unobserved word-pairs. We refer to such methods as Context-Guided Relation Embeddings (CGRE). Our proposed method differs from existing pattern-based approaches in two important ways. First, we do not require the two words to co-occur within the same sentences in a corpus to be able to represent the relation between them. Second, the parametrised operator we learn generalises in the sense that it can be applied to any new word-pair or relation type, not limited to the words and relations that exist in the training data. Empirically, relation representations obtained by our proposed CGRE model yields improvements in ranking word-pairs according to their prototypicality of the relation.

The section is structured as follows. The proposed CGRE is explained in Section 5.3.1. In Section 5.3.2, experimental setups including training datasets and implementation details are presented. Experimental results on the SemEval-2012 task 2 benchmark is presented in Section 5.3.3, which show the ability of the learnt operator to generalise to unobserved word-pairs outperforming previously proposed relational operators.

5.3.1 Context-guided Self-Supervised Relation Embeddings

Recall that our main goal is to accurately represent relations between words. We propose to learn a parametrised operator for relations that maps a word-pair to a relation embedding considering two sources of information: (a) word embeddings of related words, and (b) the contexts in which two related words co-occur. We want the learnt operator to overcome the sparseness problem in the pattern-based relation representations. Motivated by this, our objective is to create relation representations for word-pairs that do not co-occur or belong to unseen relations.

Given a set $\mathcal{D}$ of related word-pairs (a, b) along with their relation labels r, pre-trained word embeddings that represent the semantics of words, and a text corpus, we propose a method for learning δ-dimensional relation embeddings $\boldsymbol{r}_{cd} \in \mathbb{R}^{\delta}$ for an unseen word-pair (c, d). Relation labels for word-pairs can be manually annotated gold labels provided in the relational dataset such as in the case of DiffVec, Google analogies, and BATS, or can be pseudo labels generated from word-pair features as described later in this section. Figure 5.3

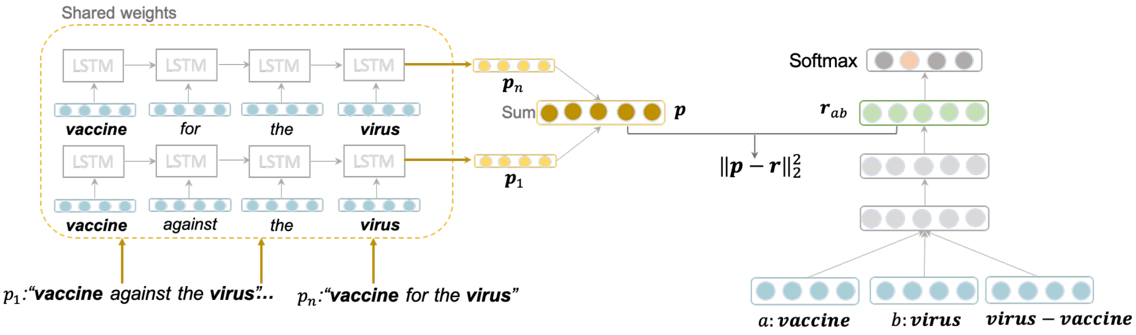


Figure 5.3: An illustration of the proposed CGRE for relation representations.

is an illustration of our proposed model. Following the prior work (Hakami and Bollegala, 2019b; Washio and Kato, 2018a,b; Joshi et al., 2019), a word-pair (a, b) is fed to a deep multilayer neural network with a nonlinearity activation for the hidden layers (as described in Section 5.2.1). In our proposed CGRE, the input layer of the network is the concatenation of embeddings $\boldsymbol{a}$ and $\boldsymbol{b}$ and their difference, $(\boldsymbol{a} \oplus \boldsymbol{b} \oplus \boldsymbol{b} - \boldsymbol{a})$. As described earlier with respect to the MnnPL, the penultimate layer of the neural network that is given by $f(a, b, \theta_f)$ is considered as a representation for a word-pair, and is passed to a fully connected softmax layer and the overall network is trained to predict the relation label for the given pair. For this purpose, we use the ℓ_2 regularised cross-entropy loss defined in (5.7) as the training objective.

$$\mathcal{J}_C = - \sum_{(a,b,r) \in \mathcal{D}} \log \quad p(r|f(a, b, \theta_f)) \tag{5.7}$$

Here, θ_f collectively denotes the parameters of the network.

$\mathcal{J}_C$ given in (5.7) does not consider the co-occurrence contexts. Therefore, we consider a relation representation operator, $g(\mathcal{P}(a, b), \theta_g)$, that encodes a set of contextual co-occurrences between a and b according to (5.8).

$$g(\mathcal{P}(a, b), \theta_g) = \sum_{p \in \mathcal{P}(a,b)} w(a, p, b) h(a, p, b, \theta_h) \tag{5.8}$$

Here, $\mathcal{P}(a, b)$ is a set of lexical patterns that co-occur with a and b. We model $h(a, p, b, \theta_h)$ by using Long Short-Term Memory (LSTM) that maps a sequences of words $w_1, w_2, \ldots, w_T$ in each pattern p (including a and b) to a fixed-length vector $\boldsymbol{p}$. The LSTM is a type of recurrent neural networks that have been proposed to encode sequential data (Hochreiter and Schmidhuber, 1997). Generally speaking, given the current input of a sequence, LSTM combines the current input representation with the previous state to generate a new hidden state that encodes the input sequence so far. The representation for a hidden state at time

t is computed as in (5.9).

$$\begin{aligned}
\boldsymbol{e}_t &= \tanh\left(\mathbf{W}_e \boldsymbol{w}_t + \mathbf{U}_e \boldsymbol{h}_{t-1} + \boldsymbol{s}_e\right) \\
\boldsymbol{i}_t &= \sigma\left(\mathbf{W}_i \boldsymbol{w}_t + \mathbf{U}_i \boldsymbol{h}_{t-1} + \boldsymbol{s}_i\right) \\
\boldsymbol{f}_t &= \sigma\left(\mathbf{W}_f \boldsymbol{w}_t + \mathbf{U}_f \boldsymbol{h}_{t-1} + \boldsymbol{s}_f\right) \\
\boldsymbol{o}_t &= \sigma\left(\mathbf{W}_o \boldsymbol{w}_t + \mathbf{U}_o \boldsymbol{h}_{t-1} + \boldsymbol{s}_o\right) \\
\boldsymbol{c}_t &= \boldsymbol{i}_t \odot \boldsymbol{e}_t + \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} \\
\boldsymbol{h}_t &= \boldsymbol{o}_t \odot \tanh\left(\boldsymbol{c}_t\right)
\end{aligned} \tag{5.9}$$

Here, $\boldsymbol{w}_t$ is the embedding of the input word w at time t, and $\boldsymbol{h}_{t-1}$ is the hidden state at the previous time step $t-1$. The initial hidden state at time 0 is typically initialised by a zero vector $\boldsymbol{h}_0 = \mathbf{0}$. The first line in (5.9) shows a standard recurrent neural network that considers $\boldsymbol{e}_t$ as a new hidden state, which obtained by a bias vector $\boldsymbol{s}_e$ and learnable weighting matrices $\mathbf{W}_e$ and $\mathbf{U}_e$ that perform an affine transformation of the current word embedding and the previous hidden state, respectively, followed by the tanh function. LSTM has additional input, forget and output gates that are respectively parametrised with $\mathbf{W}_i$, $\mathbf{W}_f$ and $\mathbf{W}_o$ to transform $\boldsymbol{w}_t$, $\mathbf{U}_i$, $\mathbf{U}_f$ and $\mathbf{U}_o$ to transform $\boldsymbol{h}_{t-1}$. σ is the sigmoid non-linear function that squashes the input to be in a range $[0, 1]$ to act as closed and open gates, and is defined as follows: $\sigma(x) = \frac{1}{1+\exp(-x)}$. The memory cell of the LSTM $\boldsymbol{c}_t$ is computed using the input gate $\boldsymbol{i}_t$ that modulates $\boldsymbol{e}_t$ and the forget gate $\boldsymbol{f}_t$ on the previous memory cell $\boldsymbol{c}_{t-1}$, where $\odot$ indicate an element-wise multiplication. Finally, the hidden state of LSTM is calculated using the output gate as in (5.9). The output at the last time step $\boldsymbol{h}_T$ is considered as a representation for the input pattern. It is worth noting that other models such as simple averaging or convolutional neural networks can also be used to encode contextual patterns, but it is not our focus here to compare all possible embeddings for patterns; rather, we aim to show the effectiveness of regularising the compositional method (i.e., $f(a, b, \theta_f)$) with contextual information. To incorporate the representativeness of a pattern for a relational instance, we assign a weight $w(a, b, p)$ given by (5.10).

$$w(a, p, b) = \frac{c(a, p, b)}{\sum_{t \in \mathcal{P}(a,b)} c(a, t, b)} \tag{5.10}$$

Here, c denotes the number of co-occurrences between p and (a, b). We have experimented with a number of normalisation strategies and found that the selected strategy works best for our settings. After encoding and weighting all the patterns in the set $\mathcal{P}(a, b)$, the sum of the pattern vectors is considered as the pattern-based relation representation for the given word-pair.

Because the holistic and compositional methods represent the same semantic relation we require them to be close in the ℓ_2 space, captured by the constraint given by (5.11).

$$\mathcal{J}_{Patt} = \frac{1}{2} \sum_{(a,b) \in \mathcal{D}} ||f(a, b, \theta_f) - g(\mathcal{P}(a, b), \theta_g)||_2^2 \tag{5.11}$$

We would like to learn word pair embeddings that simultaneously minimise both (5.7) and (5.11). Therefore, we formulate the objective function of the proposed CGRE as a linear combination of (5.7) and (5.11) as follows:

$$\mathcal{J} = \mathcal{J}_C + \lambda \mathcal{J}_{Patt} \tag{5.12}$$

Here, $\lambda \in \mathbb{R}$ is a regularisation coefficient that determines the influence of the contextual patterns of the word-pairs for the learnt relational operator. After learning CGRE, we generate a relation representation for an unseen word-pair (c, d) by concatenating $f(\boldsymbol{c}, \boldsymbol{d}, \theta_f)$ and $f(\boldsymbol{d}, \boldsymbol{c}, \theta_f)$.

Pseudo Relation Labels

To train CGRE, we require a dataset containing word-pairs annotated with relation labels. However, the cost of annotating word-pairs with relation labels can be high for specialised domains such as biomedical (Patel et al., 2018). To make our proposed method self-supervised, we induce pseudo labels for word-pairs via clustering. Specifically, we cluster the PairDiff vectors of the training word pairs using the k-means clustering algorithm with different k numbers of clusters. Because the ground truth class labels are given in DiffVec training data, we evaluate the quality of the generated clusters using the V-measure (Rosenberg and Hirschberg, 2007), which is an entropy-based measure for a harmonic mean between homogeneity and completeness of the clusters. Homogeneity (or purity) requires each cluster to contain only word-pairs of a single relation type. On the other hand, completeness is satisfied when all word-pairs of a specific relation type are assigned to the same cluster. Given a set of ground-truth classes C and a set of clusters K, the V-measure is computed as defined in (5.13).

$$\begin{aligned} &\text{Homogeneity} \quad h = 1 - \frac{H(C|K)}{H(C)} \\ &\text{Completeness} \quad c = 1 - \frac{H(K|C)}{H(K)} \\ &\text{V-measure} = 2 \times \frac{h \times c}{h + c} \end{aligned} \tag{5.13}$$

V-measure scores are between 0 (imperfect) and 1 (perfect). We examine k from 10 to 80, in steps of 10. Consistent the findings of Vylomova et al. (2016), we find that $k = 50$ clusters

Table 5.8: Examples for extracted patterns along with their weights for related word-pairs from the DiffVec training data and Wikipedia corpus.

X	Y	Pattern	Weights for patterns
cathedral	*steeple*	X*'s* Y	0.667
		X *without a* Y	0.333
vaccine	*virus*	X *for the* Y	0.429
		X *against the* Y	0.200
cottage	*wood*	X *in the* Y	0.364
		X *including a* Y	0.091
cost	*cottage*	X *of the* Y	0.333
		X *for each* Y	0.333
		X *of this* Y	0.333
hit	*rifle*	X *by* Y	0.500
		X *with* Y	0.500
radio	*battary*	X*'s* Y	0.286
		X *ran out of* Y	0.071

performs well with a V-measure of 0.416.

5.3.2 Experimental Setup

Training Data

We used the DiffVec dataset that contains $12,458$ triples (a, b, r), where words a and b are connected by an asymmetric relation r out of 36 fine-grained relation types. We use the word-pairs set $\mathcal{D}$ of the training relations and their reverse pairs to obtain relational patterns. Word-pairs in DiffVec that also appear in the test data are excluded from the training set. Following Turney (2008), we extract the context of one to five words in between the two related words considering the order in which they appear in the specified context (i.e., $\mathcal{P}(a, b)$ consists of all patterns where a occurs before b). To reduce noise, we filter out the patterns that occur between less than ten distinct word-pairs in the corpus. As a result, we obtain $5,017$ contextual patterns and the number of training triples (a, b, p) after removing out-of-vocabulary words is $158,920$. The TensorFlow-based coding, training data and pre-trained relation representations are publicly available for reproducibilitiy[6]. Examples of some obtained patterns along with their weights according to (5.10) are listed in Table 5.8. We experimented with pre-trained 300-dimensional GloVe embeddings that are

[6] `https://github.com/Huda-Hakami/Context-Guided-Relation-Embeddings`. Our implementation for the NLRA model is also included in the released GitHub repository.

trained on the Wikipedia 2014 and Gigaword (6 billion tokens)[7]. To extract co-occurrence contexts, we use the English Wikipedia corpus, which consists of ca. 337 million sentences.

Comparison methods

We compare the proposed method with unsupervised compositional operators PairDiff and Concat for the given pre-trained word embeddings. We also compare against the supervised MnnPL method proposed earlier in this chapter that learns a relation classifier using a relation labelled word-pairs and does not use contextual patterns (corresponds to $\lambda = 0$ in (5.12)).

We compare the proposed CGRE with Neural Latent Relational Analysis (NLRA) proposed by Washio and Kato (2018b). NLRA learns compositional word-pair representations from their embeddings using a feed-forward neural network, and also encode patterns using LSTM. The authors adopt CBOW-like objective function that is defined in (2.1), wherein the inner products of the compositional and pattern representations have high values for observed patterns and low values for randomly generated negative samples. Based on the contextual patterns provided by the original authors, NLRA is trained in an unsupervised fashion using all the word-pairs in the dataset (including those pairs in the test set). Because we are interested in relation representation methods that can generalise to word-pairs that *do not* co-occur in the corpus, we re-train NLRA using the same training data that we used for our proposed method such that NLRA doe not observe the word-pairs in the test dataset. The LRA relation representation method (introduced in Section 2.4.2) requires all word-pairs to be represented using lexical patterns extracted from the co-occurrence contexts. Because we strictly focus on evaluating relation representations for word-pairs without using their contextual patterns, LRA is excluded from the evaluations. Following Washio and Kato (2018b), we also evaluate the performance of each learnt relation representation method when it is combined with PairDiff. Simply, we average the scores of a learnt method and the PairDiff score for each target word-pair.

Implementation Details

For a given word-pair (a, b), we compose their embeddings $\boldsymbol{a}$ and $\boldsymbol{b}$ using a multi-layer feedforward neural networks with three hidden layers followed by the batch normalisation and the tanh nonlinearity function. Batch normalisation is a technique proposed by Ioffe and Szegedy (2015) for accelerating deep network training by reducing internal covariate shift. Word embeddings were first normalised to unit ℓ_2 length before feeding them to the neural net. The size of the hidden layers, and thus relation embeddings, are set to 300. We did not update the input word embeddings during training to preserve their distributional

[7] http://nlp.stanford.edu/data/glove.6B.zip

Table 5.9: Average MaxDiff accuracy and Spearman correlation for the 69 test relations in SemEval 2012 Task 2. Best results are in bold.

Method	**MaxDiff**	**Correlation**
PairDiff	43.48	0.31
Concat	41.67	0.29
NLRA	42.32	0.29
NLRA+PairDiff	44.35	0.33
MnnPL($\lambda = 0$)	43.75	0.31
MnnPL+PairDiff	45.42	0.35
CGRE-Gold	44.87	0.34
CGRE-Gold+PairDiff	**45.92**	**0.37**
CGRE-Proxy	44.34	0.34
CGRE-Proxy+PairDiff	45.49	0.36

regularity. A unidirectional LSTM with a 300 dimensional hidden state is used to encode the contextual patterns. AdaGrad (Duchi et al., 2011) with mini-batch of size 100 is used to learn the parameters of the proposed operator. All parameters are initialised by uniformly sampling from $[-1, +1]$ and the initial learning rate is set to 0.1. The best model was selected by early stopping using the MaxDiff accuracy on a validation set.

5.3.3 Measuring the Degrees of Prototypicality

We evaluate the relation embeddings on measuring degrees of relational similarity task using SemEval-2012 Task 2 dataset (refer to Section 3.3.2). Recall that the task is to rank word-pairs in a relation according to their degrees of prototypicality (i.e., the extent to which they exhibit the relation). Following the standard practice, we report performance on the test set (69 relations) and use train set (ten relations) for setting hyperparameters.

Table 5.9 shows the macro-averaged MaxDiff accuracy and Spearman correlations for the 69 test relations in the SemEval2012 Task 2 dataset. Our proposed method (CGRE) achieved the best results on both evaluation metrics when combined with PairDiff. CGRE trained using pseudo labels (CGRE-Proxy) can successfully reach the performance of CGRE trained using the gold labels in the DiffVec dataset (CGRE-Gold). This is encouraging because it shows that GCRE can be trained in a self-supervised manner, without requiring manually labelled data. Overall, for all the methods, adding the relational similarity scores from PairDiff improves the performance of ranking the word-pairs, which confirm the complementary properties between the two approaches when it comes to representing relations. As seen in Table 5.9, NLRA performs poorly when it is trained on DiffVec using patterns extracted for the word-pairs in DiffVec and tested on SemEval[8]. This shows that

[8] The accuracy of NLRA when its trained on pattern extracted using word pairs in the entire SemEval dataset is 45.28%, which is similar to the result reported in the original NLRA paper.

Table 5.10: Average MaxDiff (top) and Spearman correlation (bottom) for each major relation in the test set of SemEval 2012-task2. The values between parentheses indicate the performance of a method combined with PairDiff. Best results for each relation are in bold.

	MaxDiff			
Relation	PairDiff	MnnPL	CGRE-Gold	CGRE-Proxy
Class-Inclusion	48.50	**52.00** (51.60)	51.40 (51.67)	50.45 (49.35)
Part-Whole	43.50	41.33 (43.36)	39.61 (42.80)	43.35 (**44.38**)
Similar	41.26	36.20 (41.15)	40.02 (40.82)	**41.68** (41.10)
Contrast	33.72	38.57 (38.73)	**40.21** (38.44)	36.39 (36.67)
Attribute	46.32	44.84 (47.23)	46.19 (**47.97**)	45.44 (47.83)
Non-Attribute	39.11	42.45 (41.82)	42.41 (42.79)	**43.00** (41.85)
Case Relations	46.49	49.53 (49.57)	**52.04** (51.67)	49.46 (50.21)
Cause-Purpose	44.43	44.17 (46.89)	47.57 (**48.59**)	47.74 (48.17)
Spase-Time	49.48	45.53 (48.50)	48.62 (**50.21**)	45.36 (49.79)
Reference	41.92	45.94 (**47.84**)	41.32 (44.74)	41.52 (45.74)

	Correlation			
Relation	PairDiff	MnnPL	CGRE-Gold	CGRE-Proxy
Class-Inclusion	0.375	0.519 (**0.537**)	0.533 (0.516)	0.515 (0.462)
Part-Whole	0.287	0.245 (0.288)	0.228 (0.292)	0.314 (**0.321**)
Similar	0.252	0.186 (0.260)	0.245 (**0.286**)	0.280 (0.282)
Contrast	0.113	0.160 (0.202)	0.209 (**0.226**)	0.157 (0.171)
Attribute	0.410	0.351 (0.409)	0.396 (**0.444**)	0.387 (0.437)
Non-Attribute	0.209	0.264 (0.265)	0.287 (0.279)	**0.313** (0.274)
Case Relations	0.383	0.425 (0.467)	**0.475** (0.466)	0.419 (0.445)
Cause-Purpose	0.343	0.332 (0.384)	0.422 (**0.436**)	0.400 (0.404)
Spase-Time	0.422	0.373 (0.433)	0.432 (**0.455**)	0.385 (0.437)
Reference	0.303	0.323 (**0.377**)	0.212 (0.323)	0.295 (0.375)

NLRA is unable to generalise well to the relations in the SemEval dataset, not present in the DiffVec dataset.

To evaluate the performance for different relation types, we breakdown the results for the ten major relations in the 69 SemEval test set as presented in Table 5.10. By incorporating contextual patterns when training CGRE, we obtain better performance in eight out of the ten test relations in terms of MaxDiff and Spearman correlation. These improvements are statistically significant according to a paired t-test ($p < 0.01$). MnnPL reports the best accuracy and correlation for Class-Inclusion and Reference relations (either without or with the addition of PairDiff). Although the training DiffVec includes these two types of relations, the superiority of the compositional MnnPL, which does not incorporate relational patterns, is an indication of the effectiveness of using word embedding features to extract relation embeddings for these types of relations.

5.4 Summary

This chapter considered the problem of learning relation embeddings from word embeddings using parametrised operators that can be learnt from relation-labelled word-pairs. We experimentally show that the penultimate layer of a feed-forward neural network trained for classifying relation types (referred to as MnnPL) can accurately represent relations between two given words. In particular, some of the disadvantages of the popular PairDiff operator can be avoided by using MnnPL, which works consistently well for both lexicographic and encyclopaedic relations. The relation representations learnt by MnnPL generalise well to previously unseen (out-of-domain) relations, even though the number of training instances was typically small in our experiments. The analysis of near and far analogies highlighted some important limitations in the evaluation protocol used in prior work for relation composition operators. The presented work questions the belief that non-parametric operators such as PairDiff can discover rich relational structures in the word embedding space. More importantly, we show that simple supervised relational composition operators can accurately recover the relational regularities hidden inside word embedding spaces.

Prior work showed that accessing lexical relations, such as hypernym, relying only on distributional word embeddings that are trained considering 2-ways co-occurrences between words is insufficient (Roller et al., 2018). Although data sparsity is one of the main obstacles in pattern-based relation representation methods, the advantages of using contextual patterns have been proven to detect such relation types. Indeed, it is expected that the pattern-based and compositional approaches for representing relations have complementary properties. In this chapter, we sought to unify the two approaches for relation representations while overcoming the sparsity problem at the same time. We proposed CGRE, which is a method that uses the contextual patterns in a corpus to improve the compositional relation representation using word embeddings of the related word-pairs. In particular, CGRE is learnt using the contexts where two words co-occur in a corpus requiring that a pattern representation being similar to a compositional representation computed using the corresponding word embeddings. We demonstrated that by supplying a relation representation method with pattern-level information during the training improves the performance and make usage of compositional operators more efficient. Experiments on measuring degrees of relational similarity between word pairs show that we can overcome the sparsity problem of the pattern-based approaches for relations.

The next chapter will consider representing relational facts in structured KGs. The chapter will introduce a proposed relational walk generative model for KGEs. The motivation for the proposed relational walk model is to provide theoretical understanding of KGE methods.

6

Relations in Knowledge Graphs

6.1 Introduction

Earlier chapters were dedicated for using a text corpus in multiple ways as a source to infer relations between words, either by applying compositional operators on pre-trained word embeddings that are learnt using a corpus; or by incorporating contextual patterns in which related words co-occur in the corpus. Another source of information that organises relational facts is structured KGs, where entities are represented by nodes and relations between two entities are represented by the edges that connect the corresponding nodes. Nodes in KGs can be words as in the WordNet or named entities as in FreeBase or Cyc, to name a few. If we consider the nodes and the edges in a KG as words and co-occurring patterns, respectively, the KG can be seen as a corpus in which the contexts of a node are the neighbouring nodes with their relations. However, a KG is considered as a well-defined explicit source for relational facts, whereas relational information in an unstructured text is latent and based on linguistic features.

In NLP, KGs have been used widely for various tasks such as dialogue systems (Young et al., 2018; Moon et al., 2019), named entity recognition (Sui et al., 2019; López et al., 2019) and question answering (Zhang et al., 2016; Sydorova et al., 2019). The critical issue of most existing large scale KGs is that they are hard to manipulate because of sparseness (i.e., few valid links and many missing facts). As discussed earlier in Section 2.7.2, embedding methods for NLP have been spread widely from the level of words in a textual context to relations and entities in KGs. Previous work has shown that by embedding the entities and relations of a KG in some (possibly latent low-dimensional) space, we can predict links (relations) that do not exist between entities in the graph without requiring extra knowledge. This is particularly useful for expanding otherwise sparse (i.e., incomplete) KGs by reasoning in KG embedding space.

This chapter focuses on representing relations between entities in KGs. The chapter is divided into two sections as follows. In the first section, 6.2, we develop a theoretical model, Relational Walk (RelWalk), that performs a random walk over a KG to explain what latent structure is being captured by KGEs. The proposed model is an extension of the random walk model of word embeddings (Arora et al., 2016) for KGEs to derive a scoring function that evaluates the strength of a relation r between two entities h (head) and t (tail) using their embeddings. By doing so, we also propose a learning objective, motivated by theoretical analysis, to learn KGEs from a given KG. Then, in Section 6.3, the problem of representing novel relation types is considered to further combat the incompleteness problem of most existing KGs by dealing with emerging new relations. In particular, *relation composition* is introduced as the task of inferring embeddings for unseen relations (not in training data) by combining existing relations in a KG assuming that the set of relations that holds between two entities are not independent. Relation composition can be seen as an instance of the zero-shot learning setting, where the representations we compute do not correspond to any of the relations we have in the training data.

6.2 A Latent Variable Model Approach to KGEs: RelWalk

KGE can be seen as a two-step process. Given a KG represented by a set of relational triples (h, r, t), where a semantic relation r holds between a head entity h and a tail entity t, first a scoring function is defined that measures the *relational strength* of a triple (h, r, t). Second, the entity and relation embeddings in a latent semantic space that optimise the defined scoring function are learnt using some optimisation method. Despite the wide application of entity and relation embeddings created via KGE methods, the existing scoring functions (refer to Section 2.7.2) are heuristically motivated to capture some geometric requirements of the embedding space.

Despite the good empirical performance of the existing KGE methods, theoretical understanding of KGE methods is comparatively underdeveloped. For example, it is not clear how the heuristically defined KGE objectives relate to the generative process of a KG. In this work, we attempt to fill this void by providing a theoretical analysis of KGEs. Specifically, we propose a random walk generative process where we explain the formation of a relation r between two entities h and t using the corresponding relation and entity embeddings. We refer to our model as Relational Walk (abbreviated to RelWalk), where the set of all entity and relation embeddings are the latent variables for the RelWalk generative model which in turn correspond to semantics. Following this generative story, we derive a relationship between the probability of r holding between h and t, $p(h, t \mid r)$, and the embeddings of r, h and t. Interestingly, the derived relationship is not covered by any of the previously proposed heuristically-motivated scoring functions, providing the KGE method with a provable generative explanation.

The proposed RelWalk model extends the random walk analysis by Arora et al. (2016), where the authors aimed to figure out the property of a language that causes the PMI co-occurrence matrix to have approximate low rank under SVD decomposition. In particular, Arora et al. (2016) proposed a latent variable model where the words in a corpus are generated by a probabilistic model parametrised by a time-dependent discourse vector that performs a random walk. This random walk analysis derives a useful connection between the joint co-occurrence probability of two words and the ℓ_2 norm of the sum of the corresponding word embeddings. However, unlike RelWalk, they do not consider the relations between two co-occurring words in a corpus. Moreover, Bollegala et al. (2018) extended the model proposed by Arora et al. (2016) to capture co-occurrences involving more than two words. Bollegala et al. (2018) defined the co-occurrence of k unique words in a given context as a k-way co-occurrence, where Arora et al. (2016)'s result could be seen as a special case corresponding to $k = 2$. Moreover, Bollegala et al. (2018) showed that it is possible to learn word embeddings that capture some types of semantic relations such as antonymy and collocation using 3-way co-occurrences more accurately than using 2-way co-occurrences. However, their model does not explicitly consider the relations between words and uses only a corpus for learning the word embeddings.

This section of the chapter is organised as follows. We introduce the RelWalk model in Section 6.2.1. Then, in Section 6.2.2, we show that the *margin loss*, a popular objective used in much prior work in KGEs, naturally arises as the log-likelihood ratio maximisation under the probabilities estimated from the KGEs according to our theoretical relationship. In this light, we derive a training objective that we subsequently optimise for learning KGEs that empirically satisfies our theoretical relationship. Using standard benchmark datasets proposed in prior work on KGE learning, we evaluate the learnt KGEs on a link prediction and a triple classification as shown in Section 6.2.3. Experimental results show that the learnt KGEs obtain good performance on standard benchmarks for KGE methods, thereby providing empirical evidence to support the theoretical analysis of RelWalk. In Section 6.2.4, we experimentally evaluate the assumptions made for proving our theorem. Since RelWalk represents relations by matrices, low-rank approximations to the RelWalk relation embeddings have been discussed in Section 6.2.5.

6.2.1 Relational Walk

Let us consider a knowledge graph $\mathcal{D}$ where the *knowledge* is represented by relational triples $(h, r, t) \in \mathcal{D}$. Here, r is a relational predicate with two arguments, where h (*head*) and t (*tail*) entities respectively filling the first and second arguments. In this part of the work, we assume relations to be asymmetric in general. In other words, if $(h, r, t) \in \mathcal{D}$ then it does not necessarily follow that $(t, r, h) \in \mathcal{D}$. A KG can then be seen as a directed edge-labelled graph where vertices represent entities and an edge connecting two vertices represents a semantic relation that exists between the corresponding entities (Lao et al., 2011; Lao and Cohen, 2010; Gardner et al., 2013). The goal of KGE is to learn embeddings for the relations and entities in the KG such that the entities that participate in similar relations are embedded closely to each other in the entity embedding space, while at the same time relations that hold between similar entities are embedded closely to each other in the relational embedding space. We call the learnt entity and relation embeddings collectively as KGEs. Following prior work on KGEs (Bordes et al., 2011; Yang et al., 2015; Trouillon et al., 2016), we assume that entities and relations are embedded in the same vector space, allowing us to perform linear algebraic operations using the embeddings in the same vector space.

Let us consider a random walk characterised by a time-dependent *knowledge vector* $\boldsymbol{c}_k$, where k is the current time step. The knowledge vector represents the knowledge we have about a particular group of entities and relations that express some facts about the world. For example, the knowledge that we have about people that are employed by companies can be expressed using entities of classes such as people and organisation, using relations such as CEO-of, employed-at, works-for. We assume that entities h and t are represented by time-independent d-dimensional vectors, respectively $\boldsymbol{h}, \boldsymbol{t} \in \mathbb{R}^d$.

We assume the task of generating a relational triple (h, r, t) in a given KG to be a two-step process as described next. First, given the current knowledge vector at time k, $\boldsymbol{c} = \boldsymbol{c}_k$ and the relation r, we assume that the probability of an entity h satisfying the first argument of r to be given by (6.1).

$$p(h \mid r, \boldsymbol{c}) = \frac{1}{Z_c} \exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right). \tag{6.1}$$

Here, $\mathbf{R}_1 \in \mathbb{R}^{d \times d}$ is a relation-specific orthogonal matrix that evaluates the appropriateness of h for the first argument of r. For example, if r is the CEO-of relation, we would require a person as the first argument and a company as the second argument of r. However, note that the role of $\mathbf{R}_1$ extends beyond simply checking the types of the entities that can fill the first argument of a relation. For our example above, not all people are CEOs and $\mathbf{R}_1$ evaluates the likelihood of a person to be selected as the first argument of the CEO-of relation. Z_c is a normalisation coefficient such that $\sum_{h \in \mathcal{V}} p(h \mid r, \boldsymbol{c}) = 1$, where the vocabulary $\mathcal{V}$ is the set of all entities in the KG. We can use different vocabularies for the first and second arguments. However, for simplicity, we use a common vocabulary.

After generating h, the state of our random walk changes to $\boldsymbol{c}' = \boldsymbol{c}_{k+1}$, and we next generate the second argument of r with the probability given by (6.2).

$$p(t \mid r, \boldsymbol{c}') = \frac{1}{Z_{c'}} \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}'\right). \tag{6.2}$$

Here, $\mathbf{R}_2 \in \mathbb{R}^{d \times d}$ is a relation-specific orthogonal matrix that evaluates the appropriateness of

t as the second argument of r. $Z_{c'}$ is a normalisation coefficient such that $\sum_{t \in \mathcal{V}} p(t \mid r, \boldsymbol{c}') = 1$. Following our previous example of the CEO-of relation, $\mathbf{R}_2$ evaluates the likelihood of an organisation to be a company with a CEO-of position. Importantly, $\mathbf{R}_1$ and $\mathbf{R}_2$ are representations of the relation r and independent of the entities. Therefore, we consider ($\mathbf{R}_1$ and $\mathbf{R}_2$) to collectively represent the embedding of r. Orthogonality of $\mathbf{R}_1, \mathbf{R}_2$ is a requirement for the mathematical proof and also act as a regularisation constraint to prevent overfitting by restricting the relational embedding space. Intuitively, orthogonality of the relation embedding matrices ensures that the length of the head and tail entity embeddings are not altered during the generation of the tuple. In prior work, Ethayarajh (2019) shows that orthogonal matrices can represent relations.

The knowledge vector $\boldsymbol{c}_k$ performs a *slow* random walk (meaning $\boldsymbol{c}_{k+1}$ is obtained from $\boldsymbol{c}_k$ by adding a small random displacement vector) such that the head and tail entities of a relation are generated under similar knowledge vectors. More specifically, we assume that $||\boldsymbol{c}_k - \boldsymbol{c}_{k+1}|| \leq \epsilon_2$ for some small $\epsilon_2 > 0$. This is a realistic assumption for generating the two entity arguments in the same relational triple because, if the knowledge vectors were significantly different in the two generation steps, then it is likely that the corresponding relations are also different, which would not be coherent with the above-described generative process. Moreover, we assume that the knowledge vectors are distributed uniformly in the unit sphere and denote the distribution of knowledge vectors by $\mathcal{C}$.

To relate KGEs to the connections in the KG, we must estimate the probability that h and t satisfy the relation r, $p(h, t \mid r)$, which can be obtained by taking the expectation of $p(h, t \mid r, \boldsymbol{c}, \boldsymbol{c}')$ w.r.t. the two consecutive knowledge vectors $\boldsymbol{c}, \boldsymbol{c}' \sim \mathcal{C}$ given by (6.3).

$$\begin{aligned} p(h, t \mid r) &= \mathbb{E}_{\boldsymbol{c}, \boldsymbol{c}'} \left[p(h, t \mid r, \boldsymbol{c}, \boldsymbol{c}') \right] && (6.3) \\ &= \mathbb{E}_{\boldsymbol{c}, \boldsymbol{c}'} \left[p(h \mid r, \boldsymbol{c}) p(t \mid r, \boldsymbol{c}') \right] && (6.4) \\ &= \mathbb{E}_{\boldsymbol{c}, \boldsymbol{c}'} \left[\frac{\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)}{Z_c} \frac{\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}'\right)}{Z_{c'}} \right]. && (6.5) \end{aligned}$$

Here, (6.4) follows from our two-step generative process where the generation of h and t in each step is independent given the relation and the corresponding knowledge vectors. The partition functions Z_c and $Z_{c'}$ are given by:

$$Z_c = \sum_{h \in \mathcal{V}} \exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) \quad (6.6)$$

$$Z_{c'} = \sum_{t \in \mathcal{V}} \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}'\right) \quad (6.7)$$

Computing the expectation in (6.5) is generally difficult because of the two partition functions Z_c and $Z_{c'}$. However, in the next section, we show that the partition functions are narrowly distributed around a constant value for all c (or c') values with high probability.

Concentration of Partition Functions

Lemma 1 (**Concentration Lemma**). *If the entity embedding vectors satisfy the Bayesian prior $\boldsymbol{v} = s\hat{\boldsymbol{v}}$, where $\hat{\boldsymbol{v}}$ is from the spherical Gaussian distribution, and s is a scalar random variable, which is always bounded by a constant κ, then the entire ensemble of entity embeddings satisfies that:*

$$\Pr_{c \sim \mathcal{C}}[(1-\epsilon_z)Z \leq Z_c \leq (1+\epsilon_z)Z] \geq 1-\delta, \tag{6.8}$$

for $\epsilon_z = O(1/\sqrt{n})$, and $\delta = \exp(-\Omega(\log^2 n))$, where $n \geq d$ is the number of entities and Z_c is the partition function for c given by $\sum_{h \in \mathcal{V}} \exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)$.

Proof of Lemma 1: To prove the concentration lemma, we show that the mean $\mathbb{E}_{\boldsymbol{h}}[Z_c]$ of Z_c is concentrated around a constant for all knowledge vectors $\boldsymbol{c}$ and its variance is bounded.

If $\mathbf{P}$ is an orthogonal matrix and $\boldsymbol{x}$ is a vector, then $\left|\left|\mathbf{P}^\top \boldsymbol{x}\right|\right|_2^2 = (\mathbf{P}^\top \boldsymbol{x})^\top (\mathbf{P}^\top \boldsymbol{x}) = \boldsymbol{x}^\top \mathbf{P}\mathbf{P}^\top \boldsymbol{x} = ||\boldsymbol{x}||_2^2$, because $\mathbf{P}^\top \mathbf{P} = \mathbf{I}$. Therefore, from (6.6) and the orthogonality of the relational embeddings, we see that $\mathbf{R}_1 \boldsymbol{c}$ is a simple rotation of $\boldsymbol{c}$ and does not alter the length of $\boldsymbol{c}$. We represent $\boldsymbol{h} = s_h \hat{\boldsymbol{h}}$, where $s_h = ||\boldsymbol{h}||$ and $\hat{\boldsymbol{h}}$ is a unit vector (i.e., $\left|\left|\hat{\boldsymbol{h}}\right|\right|_2 = 1$) distributed on the spherical Gaussian with zero mean and unit covariance matrix $\mathbf{I}_d \in \mathbb{R}^{d \times d}$. Let s be a random variable that has the same distribution as s_h. Moreover, let us assume that s is upper bounded by a constant κ such that $s \leq \kappa$. From the assumption of the knowledge vector $\boldsymbol{c}$, it is on the unit sphere as well, which is then rotated by $\mathbf{R}_1$.

We can write the partition function using the inner-product between two vectors $\boldsymbol{h}$ and $\mathbf{R}_1 \boldsymbol{c}$, $Z_c = \sum_{h \in \mathcal{V}} \exp\left(\boldsymbol{h}^\top (\mathbf{R}_1 \boldsymbol{c})\right)$. Arora et al. (2016) showed that (Lemma 2.1 in their paper) the expectation of a partition function of this form can be approximated as follows:

$$\begin{aligned} \mathbb{E}_{\boldsymbol{h}}[Z_c] &= n\mathbb{E}_{\boldsymbol{h}}\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)\right] & (6.9) \\ &\geq n\mathbb{E}_{\boldsymbol{h}}\left[1 + \boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right] = n. & (6.10) \end{aligned}$$

where $n = |\mathcal{V}|$ is the number of entities in the vocabulary. (6.9) follows from the expectation of a sum and the independence of $\boldsymbol{h}$ and $\mathbf{R}_1$ from $\boldsymbol{c}$. The inequality of (6.10) is obtained by applying the Taylor expansion of the exponential series and the final equality is due to the symmetry of the spherical Gaussian. From the law of total expectation, we can write

$$\mathbb{E}_{\boldsymbol{h}}[Z_c] = n\mathbb{E}_{\boldsymbol{h}}\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)\right] = n\mathbb{E}_{s_h}\left[\mathbb{E}_{x|s_h}\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) \mid s_h\right]\right]. \tag{6.11}$$

where, $x = \boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}$. Note that conditioned on s_h, $\boldsymbol{h}$ is a Gaussian random variable with variance $\sigma^2 = s_h^2$. Therefore, conditioned on s_h, x is a random variable with variance

$\sigma^2 = \sigma_h^2$. Using this distribution, we can evaluate $\mathbb{E}_{x|s_h}\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)\right]$ as follows:

$$\begin{aligned}
&\mathbb{E}_{x|s_h}\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) \mid s_h\right] = \int_x \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \exp(x) dx \\
&= \int_x \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(x-\sigma^2\right)^2}{2\sigma^2} + \sigma^2/2\right) dx \\
&= \exp(\sigma^2/2). \qquad (6.12)
\end{aligned}$$

Therefore, it follows that

$$\mathbb{E}_{\boldsymbol{h}}[Z_c] = n\mathbb{E}_{s_h}\left[\exp\left(\sigma^2/2\right)\right] = n\mathbb{E}_{s_h}\left[\exp\left(s_h^2/2\right)\right] = n\exp\left(s^2/2\right), \qquad (6.13)$$

where s is the variance of the ℓ_2 norms of the entity embeddings. Because the set of entities is given and fixed, both n and σ are constants, proving that $\mathbb{E}_{\boldsymbol{h}}[Z_c]$ does not depend on c.

Next, we calculate the variance $\mathbb{V}_{\boldsymbol{c}}[Z_c]$ as follows:

$$\begin{aligned}
\mathbb{V}_{\boldsymbol{h}}[Z_c] &= \sum_h \mathbb{V}_{\boldsymbol{h}}\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)\right] \\
&\leq n\mathbb{E}_{\boldsymbol{h}}\left[\exp\left(2\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)\right] \\
&= n\mathbb{E}_{s_h}\left[\mathbb{E}_{x|s_h}\left[\exp\left(2\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) \mid s_h\right]\right]. \qquad (6.14)
\end{aligned}$$

Because $2\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}$ is a Gaussian random variable with variance $4\sigma^2 = 4s_h^2$, from a similar calculation as in (6.12) we obtain:

$$\mathbb{E}_{x|s_h}\left[\exp\left(2\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) \mid s_h\right] = \exp(2\sigma^2). \qquad (6.15)$$

By substituting (6.15) in (6.14) we have that

$$\mathbb{V}_{\boldsymbol{h}}[Z_c] \leq n\mathbb{E}_{s_h}\left[\exp\left(2\sigma^2\right)\right] = n\mathbb{E}_{s_h}\left[\exp\left(2s^2\right)\right] \leq \Lambda n \qquad (6.16)$$

for $\Lambda = \exp(8\kappa^2)$ a constant bounding $s \leq \kappa$ as stated.

From above, we have bounded both the mean and variance of the partition function by constants that are independent of the knowledge vector. Note that neither $\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)$ nor $\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)$ are sub-Gaussian nor sub-exponential. Therefore, standard concentration bounds derived for sub-Gaussian or sub-exponential random variables cannot be used in our analysis. However, the argument given in Appendix A.1 in Arora et al. (2016) for a partition function with bounded mean and variance can be directly applied to Z_c in our case, which completes the proof of the concentration lemma. From the symmetry between h and t, Lemma 1 also applies for the partition function $Z_{c'} = \sum_{t \in \mathcal{V}} \left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)$. □

Under the conditions required to satisfy Lemma 1, we proved the RelWalk theorem in the next section, which relates KGEs to the connections in the KG.

RelWalk Theorem and Proof

Theorem 2. *Suppose that the entity embeddings satisfy Lemma 1. Then, we have*

$$\log p(h, t \mid r) = \frac{\left|\left|\mathbf{R}_1{}^\top \boldsymbol{h} + \mathbf{R}_2{}^\top \boldsymbol{t}\right|\right|_2^2}{2d} - 2 \log Z \pm \epsilon. \tag{6.17}$$

for $\epsilon = O(1/\sqrt{n}) + \widetilde{O}(1/d)$*, where*

$$Z = Z_c = Z_{c'}. \tag{6.18}$$

Proof. Let us consider the probabilistic event that $(1 - \epsilon_z)Z \leq Z_c \leq (1 + \epsilon_z)Z$ to be F_c and $(1 - \epsilon_z)Z \leq Z_{c'} \leq (1 + \epsilon_z)Z$ to be $F_{c'}$. From Lemma 1 we have $\Pr[F_c] \geq 1 - \delta$. Then from the union bound we have,

$$\begin{aligned} \Pr[\bar{F}_c \cup \bar{F}_{c'}] &\leq \Pr[\bar{F}_c] + \Pr[\bar{F}_{c'}] \\ &= 1 - \Pr[F_c] + 1 - \Pr[F_{c'}] \\ &= 2\delta. \end{aligned} \tag{6.19}$$

where $\bar{F}$ is the complement of event F. Moreover, let F be the probabilistic event that both F_c and $F_{c'}$ being True. Then from $\Pr[F] = 1 - \Pr[\bar{F}_c \cup \bar{F}_{c'}]$ we have, $\Pr[F] \geq 1 - 2\exp\left(-\Omega\left(\log^2 n\right)\right)$. The R.H.S. of (6.5) can be split into two parts T_1 and T_2 according to whether F happens or not.

$$p(h, t \mid r) = \underbrace{\mathbb{E}_{\boldsymbol{c},\boldsymbol{c'}}\left[\frac{\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)}{Z_c} \frac{\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)}{Z_{c'}} \mathbf{1}_F\right]}_{T_1} + \underbrace{\mathbb{E}_{\boldsymbol{c},\boldsymbol{c'}}\left[\frac{\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)}{Z_c} \frac{\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)}{Z_{c'}} \mathbf{1}_{\bar{F}}\right]}_{T_2}. \tag{6.20}$$

Here, $\mathbf{1}_F$ and $\mathbf{1}_{\bar{F}}$ are indicator functions of the events F and $\bar{F}$ given as follows:

$$\mathbf{1}_F = \begin{cases} 1 & \text{if } F \text{ is True,} \\ 0 & \text{otherwise,} \end{cases} \text{ and } \mathbf{1}_{\bar{F}} = \begin{cases} 0 & \text{if } F \text{ is True,} \\ 1 & \text{otherwise.} \end{cases}$$

Let us first show that T_2 is negligibly small. For two real integrable functions $\psi_1(x)$ and $\psi_2(x)$ in $[a, b]$, the Cauchy-Schwarz's inequality states that

$$\left[\int_a^b \psi_1(x)\psi_2(x)dx\right]^2 \leq \int_a^b \left[\psi_1(x)\right]^2 dx \int_a^b \left[\psi_2(x)\right]^2 dx. \tag{6.21}$$

Applying (6.21) to T_2 in (6.20) we have:

$$\begin{aligned}
&\left(\mathbb{E}_{c,c'}\left[\frac{1}{Z_c Z_{c'}}\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)\mathbf{1}_{\bar{F}}\right]\right)^2 \\
&\leq \left(\mathbb{E}_{c,c'}\left[\frac{1}{Z_c^2}\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)^2 \mathbf{1}_{\bar{F}}\right]\right)\left(\mathbb{E}_{c,c'}\left[\frac{1}{Z_{c'}^2}\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)^2 \mathbf{1}_{\bar{F}}\right]\right) \\
&= \left(\mathbb{E}_{c}\left[\frac{1}{Z_c^2}\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)^2 \mathbb{E}_{c'|c}\left[\mathbf{1}_{\bar{F}}\right]\right]\right)\left(\mathbb{E}_{c'}\left[\frac{1}{Z_{c'}^2}\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)^2 \mathbb{E}_{c|c'}\left[\mathbf{1}_{\bar{F}}\right]\right]\right) \qquad (6.22)
\end{aligned}$$

Note that $Z_c \geq 1$ because Z_c is the sum of positive numbers and if $\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c} > 0$ for at least one of the $h \in \mathcal{V}$, then the total sum will be greater than 1. Therefore, by dropping Z_c term from the denominator we can further increase the first term in (6.22) as given by (6.23).

$$\mathbb{E}_{c}\left[\frac{1}{Z_c^2}\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)^2 \mathbb{E}_{c'|c}\left[\mathbf{1}_{\bar{F}}\right]\right] \leq \mathbb{E}_{c}\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)^2 \mathbb{E}_{c'|c}\left[\mathbf{1}_{\bar{F}}\right]\right] \qquad (6.23)$$

Let us split the expectation on the R.H.S. of (6.23) into two cases depending on whether $\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c} > 0$ or otherwise, indicated respectively by $\mathbf{1}_{(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c} > 0)}$ and $\mathbf{1}_{(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c} \leq 0)}$.

$$\begin{aligned}
&\mathbb{E}_{c}\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)^2 \mathbb{E}_{c'|c}\left[\mathbf{1}_{\bar{F}}\right]\right] \\
&= \mathbb{E}_{c}\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)^2 \mathbf{1}_{(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c} > 0)} \mathbb{E}_{c'|c}\left[\mathbf{1}_{\bar{F}}\right]\right] + \mathbb{E}_{c}\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)^2 \mathbf{1}_{(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c} \leq 0)} \mathbb{E}_{c'|c}\left[\mathbf{1}_{\bar{F}}\right]\right] \qquad (6.24)
\end{aligned}$$

The second term of (6.24) is upper bounded by

$$\mathbb{E}_{c,c'}\left[\mathbf{1}_{\bar{F}}\right] \leq \exp\left(-\Omega(\log^2 n)\right) \qquad (6.25)$$

The first term of (6.24) can be bounded as follows:

$$\begin{aligned}
\mathbb{E}_{c}\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)^2 \mathbf{1}_{(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c} > 0)} \mathbb{E}_{c'|c}\left[\mathbf{1}_{\bar{F}}\right]\right] &\leq \mathbb{E}_{c}\left[\exp(\alpha \boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c})^2 \mathbf{1}_{(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c} > 0)} \mathbb{E}_{c'|c}\left[\mathbf{1}_{\bar{F}}\right]\right] \\
&\leq \mathbb{E}_{c}\left[\exp(\alpha \boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c})^2 \mathbb{E}_{c'|c}\left[\mathbf{1}_{\bar{F}}\right]\right] \qquad (6.26)
\end{aligned}$$

where $\alpha > 1$. Therefore, it is sufficient to bound $\mathbb{E}_{c}\left[\exp(\alpha \boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c})^2 \mathbb{E}_{c'|c}\left[\mathbf{1}_{\bar{F}}\right]\right]$ when $||\boldsymbol{h}|| = \Omega(\sqrt{d})$.

Let us denote by z the random variable $2\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}$. Moreover, let $r(z) = \mathbb{E}_{c'|z}[\mathbf{1}_{\bar{F}}]$, which is a function of z between $[0, 1]$. We wish to upper bound $\mathbb{E}_c[\exp(z) r(z)]$. The worst-case $r(z)$ can be quantified using a continuous version of Abel's inequality (proved as Lemma A.4 in Arora et al. (2015)), we can upper bound $\mathbb{E}_c\left[\exp(z) r(z)\right]$ as follows:

$$\mathbb{E}_c\left[\exp(z) r(z)\right] \leq \mathbb{E}\left[\exp(z) \mathbf{1}_{[t,+\infty]}(z)\right] \qquad (6.27)$$

where t satisfies that $\mathbb{E}_c[\mathbf{1}_{[t,+\infty]}(z)] = \Pr[z \geq t] = \mathbb{E}_c[r(z)] \leq \exp(-\Omega(\log^2 n))$. Here, $\mathbf{1}_{[t,+\infty]}(z)$ is a function that takes the value 1 when $z \geq t$ and zero elsewhere. Then, we claim $\Pr_c[z \geq t] \leq \exp(-\Omega(\log^2 n))$ implies that $t \geq \Omega(\log^{.9} n)$.

If c was distributed as $\mathcal{N}(0, \frac{1}{d}\mathbf{I})$, this would be a simple tail bound. However, as c is distributed uniformly on the sphere, this requires special care, and the claim follows by applying the tail bound for the spherical distribution given by Lemma A.1 in (Arora et al., 2015) instead. Finally, applying Corollary A.3 in (Arora et al., 2015), we have:

$$\mathbb{E}[\exp(z)r(z)] \leq \mathbb{E}[\exp(z)\mathbf{1}_{[t,+\infty]}(z)] = \exp(-\Omega(\log^{1.8} n)) \tag{6.28}$$

From a similar argument as above we can obtain the same bound for c' as well. Therefore, T_2 in (6.20) can be upper bounded as follows:

$$\begin{aligned}
&\mathbb{E}_{c,c'}\left[\frac{1}{Z_c Z_{c'}} \exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right) \mathbf{1}_{\bar{F}}\right] \\
&\leq \left(\mathbb{E}_c\left[\frac{1}{Z_c^2} \exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)^2 \mathbb{E}_{c'|c}\left[\mathbf{1}_{\bar{F}}\right]\right]\right)^{1/2} \left(\mathbb{E}_{c'}\left[\frac{1}{Z_{c'}^2} \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)^2 \mathbb{E}_{c|c'}\left[\mathbf{1}_{\bar{F}}\right]\right]\right)^{1/2} \\
&\leq \exp(-\Omega(\log^{1.8} n))
\end{aligned} \tag{6.29}$$

Because $n = |\mathcal{V}|$, the size of the entity vocabulary, is large (ca. $n > 10^5$) in most knowledge graphs, we can ignore the T_2 term in (6.20).

Combining the above analysis of T_2 term with (6.20) we obtain an upper bound for $p(h, t \mid r)$ given by (6.30).

$$\begin{aligned}
p(h, t \mid r) &\leq (1 + \epsilon_z)^2 \frac{1}{Z^2} \mathbb{E}_{c,c'}\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right) \mathbf{1}_F\right] + |\mathcal{D}| \exp(-\Omega(\log^{1.8} n)) \\
&= (1 + \epsilon_z)^2 \frac{1}{Z^2} \mathbb{E}_{c,c'}\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)\right] + \delta_0
\end{aligned} \tag{6.30}$$

where $|\mathcal{D}|$ is the number of relational tuples (h, r, t) in the KB and $\delta_0 = |\mathcal{D}| \exp(-\Omega(\log^{1.8} n)) \leq \exp(-\Omega(\log^{1.8} n))$ by the fact that $Z \leq \exp(2\kappa)n = O(n)$, where κ is the upper bound on $\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}$ and $\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}$, which is regarded as a constant.

On the other hand, we can lower bound $p(h, t \mid r)$ as given by (6.31).

$$\begin{aligned}
p(h, t \mid r) &\geq (1 - \epsilon_z)^2 \frac{1}{Z^2} \mathbb{E}_{c,c'}\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right) \mathbf{1}_F\right] \\
&\geq (1 - \epsilon_z)^2 \frac{1}{Z^2} \mathbb{E}_{c,c'}\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)\right] - |\mathcal{D}| \exp(-\Omega(\log^{1.8} n)) \\
&\geq (1 - \epsilon_z)^2 \frac{1}{Z^2} \mathbb{E}_{c,c'}\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)\right] - \delta_0
\end{aligned} \tag{6.31}$$

Taking the logarithm of both sides, from (6.30) and (6.31), the multiplicative error translates to an additive error given by (6.32).

$$\begin{aligned}\log p(h, t \mid r) &= \log\left(\mathbb{E}_{c,c'}\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)\right] \pm \delta_0\right) - 2\log Z + 2\log(1 \pm \epsilon_z)\\ &= \log\left(\mathbb{E}_{c}\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)\mathbb{E}_{c'|c}\left[\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)\right]\right] \pm \delta_0\right) - 2\log Z + 2\log(1 \pm \epsilon_z)\\ &= \log\left(\mathbb{E}_{c}\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)A(c)\right] \pm \delta_0\right) - 2\log Z + 2\log(1 \pm \epsilon_z) \qquad (6.32)\end{aligned}$$

where $A(c) \coloneqq \mathbb{E}_{c'|c}\left[\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)\right]$.

We assumed that $\boldsymbol{c}$ and $\boldsymbol{c'}$ are on the unit sphere and $\mathbf{R}_1$ and $\mathbf{R}_2$ to be orthogonal matrices. Therefore, $\mathbf{R}_1\boldsymbol{c}$ and $\mathbf{R}_2\boldsymbol{c'}$ are also on the unit sphere. Moreover, if we let the upper bound of the ℓ_2 norm of the entity embeddings to be $\kappa'\sqrt{d}$, then we have $||\boldsymbol{h}|| \leq \kappa'\sqrt{d}$ and $||\boldsymbol{t}|| \leq \kappa'\sqrt{d}$. Therefore, we have

$$\langle \mathbf{R}_1\boldsymbol{h}, \boldsymbol{c'} - \boldsymbol{c}\rangle \leq ||\boldsymbol{h}||\,\big|\big|\boldsymbol{c} - \boldsymbol{c'}\big|\big| \leq \kappa'\sqrt{d}\,\big|\big|\boldsymbol{c} - \boldsymbol{c'}\big|\big| \qquad (6.33)$$

Then, we can upper bound $A(c)$ as follows:

$$\begin{aligned}A(c) &= \mathbb{E}_{c'|c}\left[\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)\right]\\ &= \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}\right)\mathbb{E}_{c'|c}\left[\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 (\boldsymbol{c'} - \boldsymbol{c})\right)\right]\\ &\leq \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}\right)\mathbb{E}_{c'|c}\left[\exp\left(\kappa'\sqrt{d}\,\big|\big|\boldsymbol{c'} - \boldsymbol{c}\big|\big|\right)\right]\\ &\leq (1 + \epsilon_2)\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}\right) \qquad (6.34)\end{aligned}$$

For some $\epsilon_2 > 0$. The last inequality holds because

$$\begin{aligned}\mathbb{E}_{c|c'}\left[\exp\left(\kappa'\sqrt{d}\,\big|\big|\boldsymbol{c'} - \boldsymbol{c}\big|\big|\right)\right] &= \int \exp\left(\kappa'\sqrt{d}\,\big|\big|\boldsymbol{c'} - \boldsymbol{c}\big|\big|\right) p(c'|c)dc'\\ &= \underbrace{\exp(\kappa'\sqrt{d})}_{\geq 1}\underbrace{\int \exp(\big|\big|\boldsymbol{c} - \boldsymbol{c'}\big|\big|)p(c'|c)dc'}_{\geq 1}\\ &= 1 + \epsilon_2 \qquad (6.35)\end{aligned}$$

To obtain a lower bound on $A(c)$ from the first-order Taylor approximation of $\exp(x) \geq 1 + x$ we observe that

$$\mathbb{E}_{c|c'}\left[\exp\left(\kappa'\sqrt{d}\,\big|\big|\boldsymbol{c'} - \boldsymbol{c}\big|\big|\right)\right] + \mathbb{E}_{c|c'}\left[\exp\left(-\kappa'\sqrt{d}\,\big|\big|\boldsymbol{c'} - \boldsymbol{c}\big|\big|\right)\right] \geq 2. \qquad (6.36)$$

Therefore, from our model assumptions we have

$$\mathbb{E}_{c|c'}\left[\exp\left(-\kappa'\sqrt{d}\,\big|\big|\boldsymbol{c'} - \boldsymbol{c}\big|\big|\right)\right] \geq 1 - \epsilon_2 \qquad (6.37)$$

Hence,

$$\begin{aligned} A(c) &= \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}\right) \mathbb{E}_{c'|c}\left[\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 (\boldsymbol{c}' - \boldsymbol{c})\right)\right] \\ &\geq \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}\right) \mathbb{E}_{c'|c}\left[\exp\left(-\kappa' \sqrt{d}\, \left|\left|\boldsymbol{c}' - \boldsymbol{c}\right|\right|\right)\right] \\ &\geq (1 - \epsilon_2) \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}\right) \end{aligned} \tag{6.38}$$

Therefore, from (6.35) and (6.38) we have

$$A(c) = (1 \pm \epsilon_2) \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}\right) \tag{6.39}$$

Plugging $A(c)$ back in (6.32) we obtain

$$\begin{aligned} \log p(h, t \mid r) &= \log\left(\mathbb{E}_c\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) A(c)\right] \pm \delta_0\right) - 2\log Z + 2\log(1 \pm \epsilon_z) \\ &= \log\left(\mathbb{E}_c\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)(1 \pm \epsilon_2)\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}\right)\right] \pm \delta_0\right) - 2\log Z + 2\log(1 \pm \epsilon_z) \\ &= \log\left(\mathbb{E}_c\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}\right)\right] \pm \delta_0\right) - 2\log Z + 2\log(1 \pm \epsilon_z) + \log(1 \pm \epsilon_2) \\ &= \log\left(\mathbb{E}_c\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c} + \boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}\right)\right] \pm \delta_0\right) - 2\log Z + 2\log(1 \pm \epsilon_z) + \log(1 \pm \epsilon_2) \\ &= \log\left(\mathbb{E}_c\left[\exp\left(\left(\mathbf{R}_1{}^\top \boldsymbol{h} + \mathbf{R}_2{}^\top \boldsymbol{t}\right)^\top \boldsymbol{c}\right)\right] \pm \delta_0\right) - 2\log Z + 2\log(1 \pm \epsilon_z) + \log(1 \pm \epsilon_2) \end{aligned} \tag{6.40}$$

Note that $\boldsymbol{c}$ has a uniform distribution over the unit sphere. In this case, from Lemma A.5 in Arora et al. (2016), (6.41) holds approximately.

$$\mathbb{E}_c\left[\exp\left(\left(\mathbf{R}_1{}^\top \boldsymbol{h} + \mathbf{R}_2{}^\top \boldsymbol{t}\right)^\top \boldsymbol{c}\right)\right] = (1 \pm \epsilon_3)\exp\left(\frac{\left|\left|\mathbf{R}_1{}^\top \boldsymbol{h} + \mathbf{R}_2{}^\top \boldsymbol{t}\right|\right|_2^2}{2d}\right) \tag{6.41}$$

where $\epsilon_3 = \tilde{O}(1/d)$. Plugging (6.41) in (6.40) we have that

$$\log p(h, t \mid r) = \frac{\left|\left|\mathbf{R}_1{}^\top \boldsymbol{h} + \mathbf{R}_2{}^\top \boldsymbol{t}\right|\right|_2^2}{2d} + O(\epsilon_z) + O(\epsilon_2) + O(\epsilon_3) + O(\delta_0') - 2\log Z \tag{6.42}$$

where $\delta_0' = \delta_0 \cdot \left(\mathbb{E}_c\left[\exp\left((\mathbf{R}_1{}^\top \boldsymbol{h} + \mathbf{R}_2{}^\top \boldsymbol{t})^\top \boldsymbol{c}\right)\right]\right)^{-1} = \exp(-\Omega(\log^{1.8} n))$. Therefore, δ_0' can be ignored. Note that $\epsilon_3 = \tilde{O}(1/d)$ and $\epsilon_z = \tilde{O}(1/\sqrt{n})$ by assumption. Therefore, we obtain that

$$\log p(h, t \mid r) = \frac{\left|\left|\mathbf{R}_1{}^\top \boldsymbol{h} + \mathbf{R}_2{}^\top \boldsymbol{t}\right|\right|_2^2}{2d} + O(\epsilon_z) + O(\epsilon_2) + \tilde{O}(1/d) - 2\log Z \tag{6.43}$$

□

The relationship given by (6.17) indicates that head and tail entity embeddings are first transformed respectively by $\mathbf{R}_1{}^\top$ and $\mathbf{R}_2{}^\top$, and the squared ℓ_2 norm of the sum of the transformed vectors is proportional to the probability $p(h, t \mid r)$. In the next section, we will infer a learning objective for KGEs from the proven theorem.

6.2.2 Learning KG Embeddings

In this section, we derive a training objective from Theorem 2 that we can then optimise to learn KGE. The goal is to empirically validate the theoretical result by evaluating the learnt KGEs. KGs represent information about relations between two entities in the form of *relational triples.* The joint probability $p(h, r, t)$ given by Theorem 2 is useful for determining whether a relation r exists between two given entities h and t. For example, if we know that with a high probability that r holds between h and t, then we can append (h, r, t) to the KG. The task of expanding KGs by predicting missing links between entities or relations is known as the *link prediction* problem. In particular, if we can automatically append such previously unknown knowledge to the KG, we can expand the KG and address the knowledge acquisition bottleneck.

To derive a criteria for determining whether a link must be predicted among entities and relations, let us consider a relational triple $(h, r, t) \in \mathcal{D}$ that exists in a given KG $\mathcal{D}$. We call such relational triples as *positive* triples because from the assumption it is known that r holds between h and t. On the other hand, consider a *negative* relational triple $(h', r, t') \in \bar{\mathcal{D}}$ formed by, for example, randomly perturbing a positive triple. A popular technique for generating such (pseudo) negative triples is to replace h or t with a randomly selected different instance of the same entity type. As an alternative for random perturbation, Cai and Wang (2018) proposed a method for generating negative instances using adversarial learning. Here, we are not concerned about the actual method used for generating the negative triples but assume a set of negative triples, $\bar{\mathcal{D}}$, generated using some method, to be given.

Given a positive triple $(h, r, t) \in \mathcal{D}$ and a negative triple $(h', r, t') \in \bar{\mathcal{D}}$, we would like to learn KGEs such that a higher probability is assigned to (h, r, t) than that assigned to (h', r, t'). We can formalise this requirement using the likelihood ratio given by (6.44).

$$\frac{p(h, t \mid r)}{p(h', t' \mid r)} \geq \eta \tag{6.44}$$

Here, $\eta > 1$ is a threshold that determines how higher we would like to set the probabilities for the positive triples compared to that of the negative triples. By taking the logarithm of both sides in (6.44) we obtain

$$\begin{aligned} \log p(h, t \mid r) - \log p(h', t' \mid r) &\geq \log \eta \\ \log \eta + \log p(h', t' \mid r) - \log p(h, t \mid r) &\leq 0 \end{aligned} \tag{6.45}$$

If a positive triple (h, r, t) is correctly assigned a higher probability than a negative triple $p(h', r, t')$, then the left hand side of (6.45) will be negative, indicating that there is no *loss* incurred during this classification task. Therefore, we can re-write (6.45) to obtain the *marginal loss*, a popular choice learning objective in prior work of KGEs, as shown in (6.46).

$$\begin{aligned} L(\mathcal{D}, \bar{\mathcal{D}}) &= \sum_{\substack{(h,r,t)\in\mathcal{D}\\(h',r,t')\in\bar{\mathcal{D}}}} \max\big(0, \log\eta + \log p(h', t' \mid r) - \log p(h, t \mid r)\big) \\ &= \max\left(0, 2d\log\eta + \left|\left|\mathbf{R}_1{}^\top \boldsymbol{h'} + \mathbf{R}_2{}^\top \boldsymbol{t'}\right|\right|_2^2 - \left|\left|\mathbf{R}_1{}^\top \boldsymbol{h} + \mathbf{R}_2{}^\top \boldsymbol{t}\right|\right|_2^2\right) \end{aligned} \tag{6.46}$$

We can assume $2d\log\eta$ to be the *margin* for the constraint violation.

Theorem 2 requires $\mathbf{R}_1$ and $\mathbf{R}_2$ to be orthogonal. To reflect this requirement, we add two ℓ_2 regularisation terms $\left|\left|\mathbf{R}_1{}^\top\mathbf{R}_1 - \mathbf{I}\right|\right|_2^2$ and $\left|\left|\mathbf{R}_2{}^\top\mathbf{R}_2 - \mathbf{I}\right|\right|_2^2$ with regularisation coefficients λ_1 and λ_2 to the objective function given by (6.46). In our experiments, we compute the gradients (6.46) w.r.t. each of the parameters $\boldsymbol{h}$, $\boldsymbol{t}$, $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ and use SGD for optimisation.

Learning with Multiple Negative Triples

We want to show how the marginal loss learning objective derived in Section 6.2.2 can be extended to learn from more than one negative triple per each positive triple. This formulation leads to *rank-based* loss objective used in prior work on KGE. Considering that negative triples are generated via random perturbation, it is important to consider multiple negative triples during training to better estimate the classification boundary.

Let us consider that we are given a positive triple, (h, r, t) and a set of K negative triples $\{(h'_k, r, t'_k)\}_{k=1}^K$. We would like our model to assign a probability, $p(h, t \mid r)$, to the positive triple that is higher than that assigned to any of the negative triples. This requirement can be written as (6.47).

$$p(h, t|r) \geq \max_{k=1,\ldots,K} p(h'_k, t'_k \mid r) \tag{6.47}$$

We could further require the ratio between the probability of the positive triple and maximum probability over all negative triples to be greater than a threshold $\eta \geq 1$ to make the requirement of (6.47) to be tighter.

$$\frac{p(h, t \mid r)}{\max_{k=1,\ldots,K} p(h'_k, t'_k \mid r)} \geq \eta \tag{6.48}$$

By taking the logarithm of (6.48) we obtain

$$\log p\left(h, t \mid r\right) - \log\left(\max_{k=1,\ldots,K} p\left(h'_k, t'_k \mid r\right)\right) \geq \log(\eta) \tag{6.49}$$

Therefore, we can define the marginal loss for a misclassification as follows:

$$L\left((h,r,t),\{(h'_k,r,t'_k)\}_{k=1}^{K}\right) = \max\left(0, \log\left(\max_{k=1,\ldots,K} p(h'_k,t'_k \mid r)\right) + \log(\eta) - \log p(h,t \mid r)\right) \tag{6.50}$$

However, from the monotonicity of the logarithm we have $\forall x_1, x_2 > 0$, if $\log(x_1) \geq \log(x_2)$ then $x_1 \geq x_2$. Therefore, the logarithm of the maximum can be replaced by the maximum of the logarithms in (6.50) as shown in (6.51).

$$L\left((h,r,t),\{(h'_k,r,t'_k)\}_{k=1}^{K}\right) = \max\left(0, \max_{k=1,\ldots,K} \log\left(p\left(h'_k,t'_k \mid r\right)\right) + \log(\eta) - \log p(h,t \mid r)\right) \tag{6.51}$$

By substituting (6.17) for the probabilities in (6.51) we obtain the rank-based loss given by (6.52).

$$\begin{aligned} &L\left((h,r,t),\{(h'_k,r,t'_k)\}_{k=1}^{K}\right) \\ &= \max\left(0, 2d\log(\eta) + \max_{k=1,\ldots,K} \left|\left|\mathbf{R}_1{}^\top \boldsymbol{h}'_k + \mathbf{R}_2{}^\top \boldsymbol{t}'_k\right|\right|_2^2 - \left|\left|\mathbf{R}_1{}^\top \boldsymbol{h} + \mathbf{R}_2{}^\top \boldsymbol{t}\right|\right|_2^2\right) \end{aligned} \tag{6.52}$$

In practice, we can use $p(h'_k, t'_k \mid r)$ to select the negative triple with the highest probability for training with the positive triple. The next section will empirically assess entity and relation embeddings of a given KG that are learnt under the scoring function derived by the RelWalk model.

6.2.3 Empirical Evaluation of RelWalk Embeddings

To empirically evaluate the theoretical result stated in Theorem 2, we learn KGEs (using the proposed RelWalk approach) by minimising the marginal loss objective derived in Section 6.2.2 considering a given KG. We generate negative triples by replacing a head or a tail entity in a positive triple by a randomly selected entity and learn KGEs. The model is trained until convergence or at most 1000 epochs over the training data where each epoch is divided into 100 mini-batches. The best model is selected by early stopping based on the performance of the learnt embeddings on the validation set (evaluated after each 20 epochs). We selected the initial learning rate (α) for SGD in $\{0.01, 0.001\}$, the regularisation coefficients (λ_1, λ_2) for the orthogonality constraints of relation matrices in $\{0, 1, 10, 100\}$. The number of randomly generated negative triples n_{neg} for each positive example is varied in $\{1, 10, 20, 50, 100\}$ and $d \in \{50, 100\}$. The RelWalk is implemented based on the open-source toolkit OpenKE[1] (Han et al., 2018). Source code and pre-trained

[1] `https://github.com/thunlp/OpenKE/tree/OpenKE-Tensorflow1.0`

embeddings for considered KGs are publicly available[2]. We conduct two evaluation tasks: *link prediction* and *triple classification*, as presented in the next sections.

Link Prediction

Link prediction is a task of predicting the missing head or tail entity in a given triple ($(?\,, r, t)$ or $(h, r, ?\,)$ (Bordes et al., 2011). We use the FB15K-237 (a subset of *Freebase* FB15K) and WN18RR (a subset of *WordNet* WN18) datasets, which are standard benchmarks for KGE methods (see Table 6.1). It has been found that FB15K and WN18 suffer from the existence of inverse relations wherein a large number of test triples can be obtained by inverting triples in the training data. To avoid such a flaw, Toutanova et al. (2015) introduced FB15K-237[3], and Dettmers et al. (2018) introduced WN18RR[4], where inverse relations in the original KGs are deleted. Since then, proposed methods for KGEs are being evaluated using the amended KGs. We use the standard training, validation and test split as detailed in Table 6.1. Optimal hyper-parameter settings on validation sets were: $d = 100$, $\lambda_1 = \lambda_2 = 10$, $n_{\text{neg}} = 20$ for FB15K-237 and 100 for WN18RR, $\alpha = 0.001$ for FB15K-237 and 0.01 for WN18RR.

Following previous studies, the performance is evaluated using three metrics, namely, Mean Reciprocal Rank (MRR), Mean Rank (MR) and hits at ranks k(H@k). MR is the average of the rank assigned to the original head or tail entity in a corrupted triple (the lower is better), whereas MRR is the average of the reciprocal ranks (the higher is better). On the other hand, H@k is the proportion of correct entities that have been ranked among the top k candidates ($k = 1, 3, 10$). We only report scores under the *filtered* setting, which removes all triples appeared in training, validating and testing sets from candidate triples before obtaining the rank of the ground truth triple (Bordes et al., 2013). During the corruption process, we consider all entities that appear in the corresponding argument in the entire KG as candidates (known as type constraint setting). Under filtering and type constraint settings, the set of head corruptions from test triples (h, r, t) can be formally defined as in (6.53). We similarly generate the tail corrupted triples.

$$\Big\{(h', r, t) \mid h' \in \{h \in \mathcal{V} \mid \exists t : (h, r, t) \in \mathcal{D}\} \wedge (h', r, t) \notin \mathcal{D}\Big\} \tag{6.53}$$

We compare the KGEs learnt by RelWalk against prior work using the published results for link prediction as shown in Table 6.2. We see that RelWalk obtains competitive performance on both WN18RR and FB15K237 under all the evaluation measures. In particular, it is outperformed only by the current state-of-the-art KGE method proposed by Lacroix et al. (2018) (CP-N3), which uses nuclear 3-norm regularisers with canonical

[2] Will be released upon paper acceptance to facilitate the double blind policy.

[3] `https://www.microsoft.com/en-us/download/details.aspx?id=52312`

[4] `https://github.com/TimDettmers/ConvE/blob/master/WN18RR.tar.gz`

Table 6.1: Statistics of the KGs used in this study.

Dataset	Relations	Entities	Train	Test	Validation
FB15K-237	237	14,541	272,115	17,535	20,466
WN18RR	11	40,943	86,835	3,134	3,034
WN11	11	38,588	112,581	10,544	2,609
FB13	13	75,043	316,232	23,733	5,908

Table 6.2: Results of Link prediction. Results marked with [$\star$] are taken from (Dettmers et al., 2018), [$\bullet$] from (Nguyen et al., 2016), [$\triangleleft$] from and (Cai and Wang, 2018). All other results for the baselines are taken from their original papers.

	FB15K237					WN18RR				
Method	MRR	MR	H@1	H@3	H@10	MRR	MR	H@1	H@3	H@10
TransE$^{\bullet}$	0.294	347	-	-	0.465	0.226	3384	-	-	0.50
TransD$^{\triangleleft}$	0.280	-	-	-	0.453	-	-	-	-	0.43
DistMult*	0.241	254	0.155	0.263	0.419	0.43	5110	0.39	0.44	0.49
ComplEx*	0.247	339	0.158	0.275	0.428	0.44	5261	0.41	0.46	0.51
ConvE	0.325	244	0.237	0.356	0.501	0.430	4187	0.40	0.44	0.52
CP-N3	**0.360**	-	-	-	**0.540**	**0.47**	-	-	-	**0.54**
RelWalk	0.329	**105**	**0.243**	**0.354**	0.502	0.451	**3232**	**0.42**	**0.47**	0.51

tensor decomposition. RelWalk's consistent good performance on both versions of this dataset shows that it is considering the global structure in the KG when learning KGEs.

Triple Classification

Triple classification is the task of predicting whether a relation r holds between h and t in a given triple (h, r, t). This binary triple classification task for evaluating KGEs has been established by Socher et al. (2013a), who released the FB13 and WN11[5] datasets with the standard splits as shown in Table 6.1. Optimal hyper-parameter settings were: $d = 100$, $\lambda_1 = \lambda_2 = 10$, $n_{\text{neg}} = 50$ for FB13 and 20 for WN11, $\alpha = 0.001$ for FB13 and 0.01 for WN11. Socher et al. (2013a) sampled a negative triple for each triple (h, r, t) in the test split of a KG by switching h or t under the type constraint setting, which considered entities that appeared in the corresponding argument in the KG. For example, (*Thomas Elyot*, profession, *philologist*) is a negative example of (*Thomas Elyot*, profession, *lexicographer*). Following Socher et al. (2013a), each relation r is assigned a threshold T_r to conduct the classification such that if $p(h, r, t) \geq T_r$ we predict (h, r, t) as a positive relation, otherwise it is labelled as negative. The validation split is used to find the appropriate thresholds for

[5] `https://cs.stanford.edu/~danqi/data/nips13-dataset.tar.bz2`

Table 6.3: Results of Triple classification. Best results are in bold.

	Accuracy	
Method	WN11	FB13
SE	53.0	75.2
TransE	75.9	81.5
TransR	85.9	82.5
TransG	**87.4**	87.3
NTN	70.4	87.1
RelWalk	76.3	**88.6**

the relations. We report the percentage of the correctly classified test triples (i.e., accuracy) as the evaluation metric for this task.

Table 6.3 compares the accuracy of different KGE methods for relation triple classification. As shown in the table, RelWalk reports the best performance on FB13, whereas TransG (Xiao et al., 2016) reports the best performance on WN11. TransG is a generative model based on the Chinese restaurant process to model multiple semantics of relations, however, the relation embeddings are designed to satisfy vector translation similar to TransE. Considering that both TransG and RelWalk are generative models, it would be interesting to further investigate generative approaches for KGE in the future. Overall, the experimental results support our theoretical claim and emphasise the importance of theoretically motivating the scoring function design process.

6.2.4 Validity of Key Assumptions

Our theoretical analysis depends on two main assumptions: (a) concentration of the partition function Z_c (Lemma 1), and (b) the orthogonality of the relation embedding matrices $\mathbf{R}_1, \mathbf{R}_2$. In this section, we empirically study the relationship between these assumptions and the performance of the RelWalk embeddings.

Given $\mathbf{R}_1$ and $\mathbf{R}_2$ learnt by RelWalk for a particular r, we can measure the degree to which the orthogonality, ν_r, is satisfied by the sum of the non-diagonal elements as given in (6.54).

$$\nu_r = \sum_{i \neq j} |{\mathbf{R}_1}^\top \mathbf{R}_1|_{ij} + |{\mathbf{R}_2}^\top \mathbf{R}_2|_{ij} \tag{6.54}$$

If a matrix $\mathbf{A}$ is orthogonal, then the non-diagonal elements of the inner-product $\mathbf{A}^\top \mathbf{A}$ will be zeros. Therefore, the smaller the ν_r values, the more orthogonal the relation embeddings are. The values of ν_r are measured for the 11 relation types in the WN18RR dataset as shown in Table 6.4. From the table, we see that ν_r values are indeed small for different

Table 6.4: Empirical analysis of the concentration of the partitioning functions and the orthogonality of the relation embeddings, and their Pearson correlation coefficients against H@10 for the relations in WN18RR dataset.

Relation	#tuples	H@10	ν_r	σ_c	$\sigma_{c'}$	$\sqrt{\sigma_c^2 + \sigma_{c'}^2}$
hypernym	1251	0.188	3.249	68.89	64.41	94.31
derivationally_related_form	1074	0.955	1.690	63.44	65.33	91.07
instance_hypernym	122	0.541	0.362	63.11	64.56	90.28
also_see	56	0.670	0.234	70.76	61.51	93.76
member_meronym	253	0.281	4.389	63.78	66.09	91.84
synset_domain_topic_of	114	0.513	0.727	65.66	65.48	92.73
has_part	172	0.247	0.548	66.21	66.50	93.84
member_of_domain_usage	24	0.688	0.045	65.24	63.16	90.81
member_of_domain_region	26	0.442	0.065	67.53	66.31	94.64
verb_group	39	0.974	0.038	64.22	63.19	90.09
similar_to	3	1.000	0.111	63.67	63.96	90.25
Correlations			-0.515	-0.392	-0.496	-0.700

relation types indicating that the orthogonality requirement is satisfied as expected. ν_r reports the lowest score for verb-group relation that achieves the best H@10 accuracy of 0.974. Interestingly, a moderately high (-0.515) negative Pearson correlation between H@10 and ν_r shows that orthogonality correlates with the better the performance. To visualise how the orthogonality affects different relation types, the elements in $\mathbf{R}_1{}^\top\mathbf{R}_1$ and $\mathbf{R}_2{}^\top\mathbf{R}_2$ are plotted for four relations in the WN18RR dataset in Figure 6.1 for 100 × 100 dimensional relational embeddings. For the two relations also_see and similar_to we see that the corresponding inner-products are sparse except in the main diagonal, compared to that in hypernym and member_meronym relations. On the other hand, according to Table 6.4 the H@10 values for also_see and similar_to are higher than that for hypernym and member_meronym as implied by the negative correlation.

To test for the concentration of the partition functions, for a relation r we compute Z_c and $Z_{c'}$ values using respectively (6.6) and (6.7) over a set of randomly sampled 10,000 head or tail entities as the knowledge vectors $\boldsymbol{c}$ (or $\boldsymbol{c}'$) ℓ_2 normalised to unit length. We compute the standard deviations σ_c and $\sigma_{c'}$ respectively for the distributions of Z_c and $Z_{c'}$ and their geometric mean as shown in Table 6.4. We observed a Gaussian-like distributions for the partition functions for different relations and smaller standard deviations indicate stronger concentration around the mean. Interestingly, from Table 6.4 we see a strong negative correlation between H@10 and the standard deviations (-0.7) indicating that the performance of RelWalk depends on the validity of the concentration assumption.

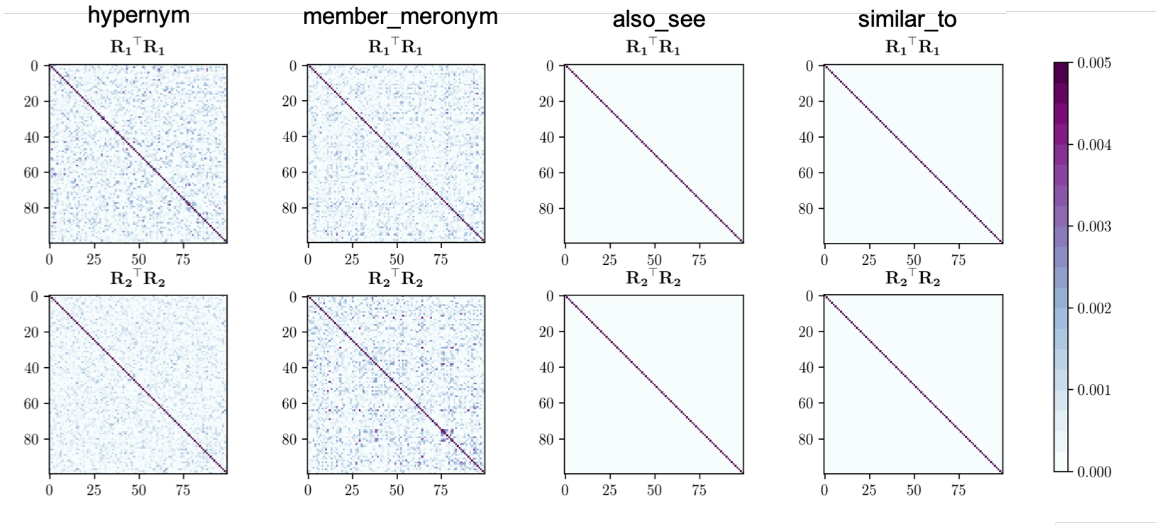


Figure 6.1: Heatmap visualisation of the orthogonality in different relation embeddings from the WN18RR.

6.2.5 Compression of Relation Embeddings

RelWalk uses (orthogonal) matrices to represent relations, which require more parameters compared to vector representations of relations. Prior work studying lower-rank decomposition of KGEs has shown that, although linear embeddings of graphs can require prohibitively large dimensionality to model certain types of relations (Nickel et al., 2014) (e.g., sameAs), nonlinear embeddings can mitigate this problem (Bouchard et al., 2015). In this section, we propose memory-efficient low-rank approximations to the proposed RelWalk relation embeddings.

From the definition of orthogonality, it follows that the relation embeddings $\mathbf{R}_1, \mathbf{R}_2 \in \mathbb{R}^{d\times d}$ learnt by RelWalk for a particular relation r are both full-rank and cannot be factorised as the product of two lower rank matrices. This prevents us from directly applying matrix decomposition methods such as non-negative matrix factorisation on relation embeddings to obtain low-rank approximations. Therefore, we subtract the identity matrix $\mathbf{I} \in \mathbb{R}^{d\times d}$ from the relation embedding $\mathbf{R}(\in \{\mathbf{R}_1, \mathbf{R}_2\})$ and factorise the remainder $\mathbf{R}' \in \mathbb{R}^{d\times d}$ as the product of two low-rank matrices using the eigendecomposition of $\mathbf{R}'$ as given by (6.55).

$$\begin{aligned}\mathbf{R} &= \mathbf{I} + \mathbf{R}' \\ &= \mathbf{I} + \mathbf{U}_R\mathbf{D}\mathbf{U}_R{}^\top \\ &\approx \mathbf{I} + \sum_{k=1}^{K} \mathbf{D}_{(k,k)}\mathbf{U}_{R(k,:)}\mathbf{U}_{R(:,k)}\end{aligned} \tag{6.55}$$

Here, $\mathbf{U}$ is the matrix formed by arranging the eigenvectors of $\mathbf{R}'$ as columns, and $\mathbf{D}$ is a

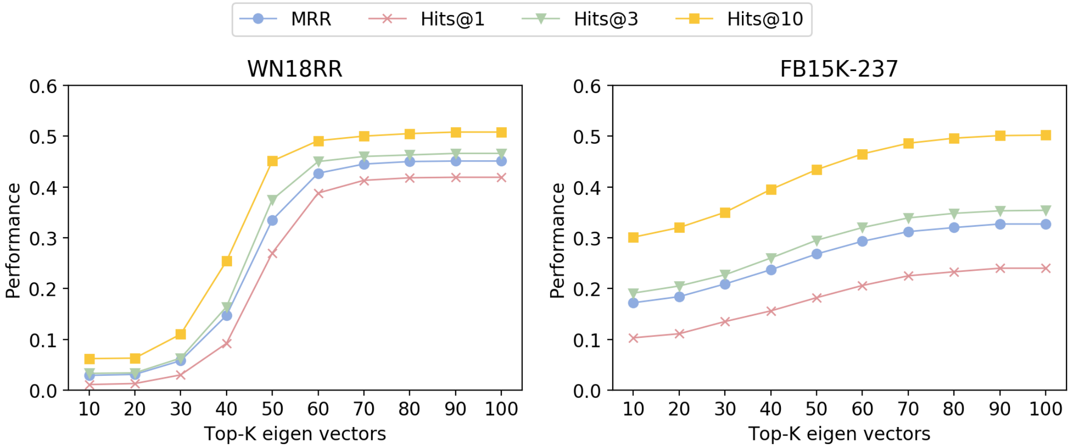


Figure 6.2: Results for the approximated relation embeddings for link prediction on WN18RR and FB15K-237.

diagonal matrix containing the eigenvalues of $\mathbf{R}'$ in the descending order.[6] We can then use the largest $K \leq d$ eigenvalues and corresponding eigenvectors to obtain a rank-K approximation in the sense of minimum Frobenius distance between $\mathbf{R}'$ and its rank-K approximation. In the case we use K factors in the approximation, we must store dK real numbers corresponding to the d-dimensional eigen vectors per each of the K components as opposed to d^2 real numbers in $\mathbf{R}$.[7] The compression ratio in this case becomes $dK/d^2 = K/d$. When $K << d$, this results in a significant compression.

To empirically evaluate the trade-off between the number of eigen vectors used in the compression and the accuracy of the learnt relation embeddings, we use the approximated relation embeddings for link prediction on WN18RR and FB15K-237 as shown respectively in the left and right plots in Figure 6.2. We use $d = 100$ dimensional relation embeddings learnt by RelWalk and we approximate using top-K eigenvectors. From Figure 6.2, we see for $K > 60$ components the performance saturates in both datasets. On the other hand, we need at least $K = 30$ components to get any meaningful accuracy for link prediction on these two datasets. With $K = 60$ and $d = 100$ this approximation results in an 60% compression ratio.

In the next section, we move to consider the problem of inferring relational embeddings for unseen relation types, where we propose a supervised relation composition method that composes existing relation embeddings.

[6] Although $\mathbf{R}'$ is square it is not symmetric. Therefore, some of the eigenvalues of $\mathbf{R}'$ can be complex in general. The absolute values are used to sort the eigenvalues in the descending order in $\mathbf{D}$.

[7] We can scale each eigen vector by the square root of the corresponding eigen value as a pre-processing step, thereby avoiding the need to store the K eigen values.

6.3 Learning to Compose Relational Embeddings in Knowledge Graphs

Let us recall that KGE methods learn lower-dimensional representations for entities and relations in KGs, which can be used to infer previously unobserved links (relations) between pairs of entities in the KG. However, the relation types that can be predicted using KGEs are confined to $\mathcal{R}$, the set of relation types that *already exists* in the KG. Although KGEs can predict links of relations that currently do not exist between two entities in the KG, these links are limited to the relation types that exist in the training data. In other words, using the pre-trained KGEs alone, we cannot predict representations for previously unseen (no in training data) relations that are encountered during test time. On the other hand, the relations that exist in a KG are often closely related (Takahashi et al., 2018). For example, given the embeddings for the relations country-of-film and currency-of-country, we can compose the embedding for a previously unseen relation such as currency-of-film_budget because entities are shared across many tuples such as (Movie, country-of-film, Country), (Country, currency-of-country, Currency). In this example, Movie, Country, and Currency can be replaced respectively by valid entities such as *The Italian Job*, *UK* and *GBP*.

To address the aforementioned issue, we propose a relation composition as a task of inferring relation embeddings for novel relation types by composing pre-trained embeddings for existing relation types. The proposed method of relation composition assumes the availability of compositional constraints for relations that contains rules of two relations $r_A \wedge r_B$ imply a third one r_C. Our problem setting differs from that of KGE methods in two important ways. First, we do not learn relation embeddings from scratch for a given KG, but instead use pre-trained KGEs and learn a composition operator to predict the embeddings for the relations that currently do not exist in the KG. Relations are fixed with pre-trained representations because during the inference time we will generate embeddings for never seen relations and hence their representations never get updated. Second, the composition functions we learn are *universal* (Riedel et al., 2013) in the sense that they are not parametrised by the entities or relations in the KG, thereby making the composition function independent from a particular KG. This is attractive because, theoretically the learnt composition function can be used to compose *any* relation type, not limited to the relations that exist in the KG used for training.

Our goal of handling new relation types in a KG is similar to that for TransW model proposed by Ma et al. (2019), which learns a mapping from word embedding space to knowledge graph space to deal with new relations and entities. Unlike our approach, TransW does not take into account the relatedness between relation types that can be inferred from shared entities across the given facts. In particular, our proposed relation composition is learnt in a supervised fashion considering a list of compositional constraints that connect relations in the form: $r_A \wedge r_B \rightarrow r_C$. We adopt global compositional constraints introduced by Takahashi et al. (2018), which are obtained from multi-hop links over a KG, as will be described later.

A number of studies have involved multi-hop relational paths between entities (i.e., $h \xrightarrow{r_A} e_1 \xrightarrow{r_B} t$) to improve the KGE methods only trained on direct links (Lin et al., 2015; Nathani et al., 2019). However, while the existing path-based KGE methods boost the performance of link prediction, they still cannot generalise over unseen relation types. A notable exception is a zero-shot relational learning setting proposed in Neelakantan et al. (2015) study under recurrent neural networks to compose relational paths. The authors learn a global composition function considering paths of all related entity-pairs in a KG which: (a) increases the complexity of the model, (b) suffers from noise since not every path is predictive for a relation in an entity-pair, and (c) based on local-statistics, i.e., individual pair-paths data. However, our proposed model makes use of global-statistic compositional constraints, which is more robust to noise as it is necessary "to zoom out" to consider the entire KG facts and collect such constraints.

Guu et al. (2015) considered path queries in a KG connecting two entities and proposed a composition method that multiplies the relation embedding matrices corresponding to the relations along the connecting path. They considered relation composition under the TransE model (Bordes et al., 2013), where relational embedding vectors are added, and under the DistMult model (Yang et al., 2015), where relations are represented using diagonal matrices. These composition operators can be seen as *unsupervised* in the sense that there are no learnable parameters in the composition function. In our experiments, we use both matrix addition and multiplication as unsupervised baseline methods for comparisons. On the other hand, our proposed method is a supervised relation composition method and we consider relations represented by orthogonal matrices, which are not diagonal in general.

The rest of this section is structured as follows. In Section 6.3.1, the proposed method for relation composition is introduced, followed by experimental settings and datasets in Section 6.3.2. Finally, empirical results are presented in 6.3.3.

6.3.1 Relation Composition

The proposed relation composition model assumes the availability of compositional constraints and pre-trained relation embeddings. Our proposed method is agnostic to the algorithm used to learn the input KGEs. In this regard, it can be used to compose relation embeddings using KGEs produced by any KGE learning method. In our experiments, we use relation embeddings learnt using RelWalk, which represent relations using matrices and report good performance on KGE benchmarks as seen in Section 6.2.3. The benefits of considering relation composition for RelWalk embeddings is that composing matrices is more computationally complex and it is also more general than composing vectorial relation embeddings (i.e., diagonal matrices can be used to represent vectors).

Let us assume that the two relations r_A and r_B jointly imply a third relation r_C, we use the notation $r_A \wedge r_B \rightarrow r_C$ to express this fact. Moreover, let us assume that the relational embeddings produced by RelWalk for r_A and r_B to be respectively $(\mathbf{R}_1^A, \mathbf{R}_2^A)$ and $(\mathbf{R}_1^B, \mathbf{R}_2^B)$. For simplify the explanation, let us assume all relation embedding matrices are in $\mathbb{R}^{d \times d}$. We model the problem of composing the relation embeddings $(\hat{\mathbf{R}}_1^C, \hat{\mathbf{R}}_2^C)$ for r_C as learning two joint compositional operators (ϕ_1, ϕ_2) such that:

$$\phi_1 : \mathbf{R}_1^A, \mathbf{R}_2^A, \mathbf{R}_1^B, \mathbf{R}_2^B \longrightarrow \hat{\mathbf{R}}_1^C \tag{6.56}$$

$$\phi_2 : \mathbf{R}_1^A, \mathbf{R}_2^A, \mathbf{R}_1^B, \mathbf{R}_2^B \longrightarrow \hat{\mathbf{R}}_2^C \tag{6.57}$$

We will first present unsupervised compositional operators for ϕ_1 and ϕ_2 before moving to introduce the proposed supervised relation composition.

Unsupervised Relation Composition

When the compositional operators ϕ_1, ϕ_2 do not have learnable parameters we call them *unsupervised*. In the case of matrix relation embeddings as in RelWalk, we consider the following unsupervised operators.

Addition:

$$\mathbf{R}_1^A + \mathbf{R}_1^B = \hat{\mathbf{R}}_1^C$$
$$\mathbf{R}_2^A + \mathbf{R}_2^B = \hat{\mathbf{R}}_2^C$$

Matrix Product:

$$\mathbf{R}_1^A \mathbf{R}_1^B = \hat{\mathbf{R}}_1^C$$
$$\mathbf{R}_2^A \mathbf{R}_2^B = \hat{\mathbf{R}}_2^C$$

Hadamard Product:

$$\mathbf{R}_1^A \odot \mathbf{R}_1^B = \hat{\mathbf{R}}_1^C$$
$$\mathbf{R}_2^A \odot \mathbf{R}_2^B = \hat{\mathbf{R}}_2^C$$

Here, $\odot$ denotes the Hadamard (elementwise) product of two matrices. Unlike the matrix product, both addition and Hadamard product are commutative.

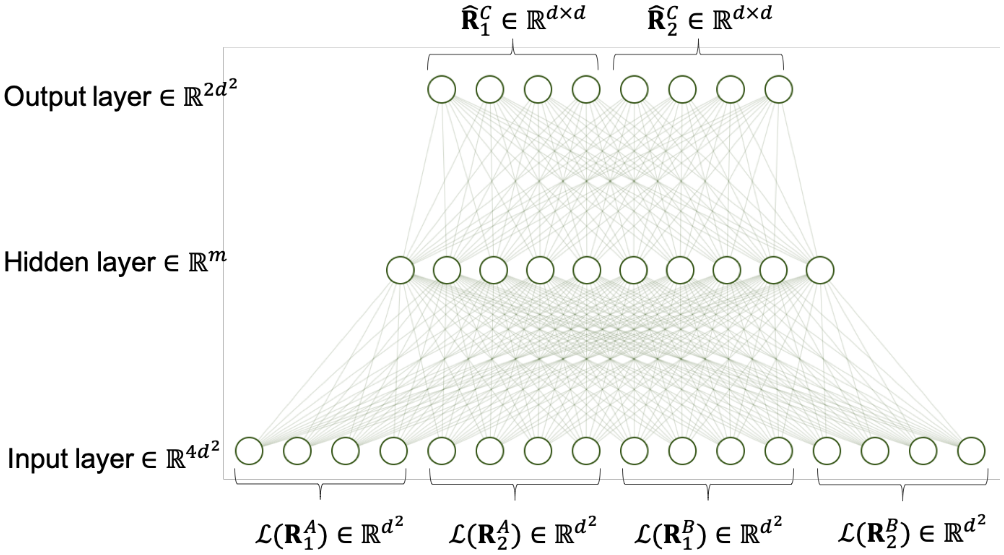

$\hat{\mathbf{R}}_1^C \in \mathbb{R}^{d\times d}$
$\hat{\mathbf{R}}_2^C \in \mathbb{R}^{d\times d}$
Output layer $\in \mathbb{R}^{2d^2}$
Hidden layer $\in \mathbb{R}^m$
Input layer $\in \mathbb{R}^{4d^2}$
$\mathcal{L}(\mathbf{R}_1^A) \in \mathbb{R}^{d^2}$
$\mathcal{L}(\mathbf{R}_2^A) \in \mathbb{R}^{d^2}$
$\mathcal{L}(\mathbf{R}_1^B) \in \mathbb{R}^{d^2}$
$\mathcal{L}(\mathbf{R}_2^B) \in \mathbb{R}^{d^2}$

Figure 6.3: An illustration of the proposed relational compositional model using RelWalk relation embeddings.

Supervised Relation Composition

The unsupervised compositional operators described above are not guaranteed to correctly predict the embeddings because they cannot be tuned to the relations in a given KG. Moreover, each unsupervised operator considers either one of $\mathbf{R}_1$ or $\mathbf{R}_2$, and do not model their possible interactions. Therefore, we propose to learn two *supervised* relation composition operators with shared parameters. The parameter sharing enables the two operators to learn a consistent relation embedding.

Different models can be used to express ϕ_1 and ϕ_2. In our work, we use feed-forward neural nets, which are universal approximators for this purpose (Hornik et al., 1989). The proposed model for predicting relation embeddings is depicted in Figure 6.3. In detail, we first linearise the input $d \times d$ matrix relation embeddings to d^2-dimensional vector embeddings via a linearisation operator $\mathfrak{L}$. We then concatenate the four linearised relational embeddings $\mathfrak{L}(\mathbf{R}_1^A), \mathfrak{L}(\mathbf{R}_2^A), \mathfrak{L}(\mathbf{R}_1^B), \mathfrak{L}(\mathbf{R}_2^B)$ and feed it to the neural network. The weight and bias for the first layer are respectively $\mathbf{W}_1 \in \mathbb{R}^{4d^2 \times m}$ and $\boldsymbol{s}_1 \in \mathbb{R}^m$, where m is the number of neurones in the hidden layer. A nonlinear activation function is applied at the hidden layer. In our experiments, we used tanh as the activation function. The weight and bias for the output layer, respectively $\mathbf{W}_2 \in \mathbb{R}^{m \times 2d^2}$ and $\boldsymbol{s}_2 \in \mathbb{R}^{2d^2}$, are chosen such that by appropriately splitting the output into two parts and applying the inverse mapping of the linearisation, we can predict $\hat{\mathbf{R}}_1^C$ and $\hat{\mathbf{R}}_2^C$. Denoting the concatenation by $\oplus$ and inverse linearisation by $\mathfrak{L}^{-1}$, we can write the predicted embeddings for r_C as follows:

$$\boldsymbol{x} = \mathfrak{L}(\mathbf{R}_1^A) \oplus \mathfrak{L}(\mathbf{R}_2^A) \oplus \mathfrak{L}(\mathbf{R}_1^B) \oplus \mathfrak{L}(\mathbf{R}_2^B) \tag{6.58}$$

$$\boldsymbol{h} = \tanh(\mathbf{W}_1\boldsymbol{x} + \boldsymbol{s}_1) \tag{6.59}$$

$$\boldsymbol{y} = \mathbf{W}_2\boldsymbol{h} + \boldsymbol{s}_2 \tag{6.60}$$

$$\hat{\mathbf{R}}_1^C = \mathfrak{L}^{-1}\boldsymbol{y}_{:d^2} \tag{6.61}$$

$$\hat{\mathbf{R}}_2^C = \mathfrak{L}^{-1}\boldsymbol{y}_{d^2:} \tag{6.62}$$

Having being provided with a training set of relational tuples $\{(r_A, r_B, r_C)\}$, where $r_A \wedge r_B \rightarrow r_C$ and their RelWalk embeddings, using Adam (Kingma and Ba, 2015), we find the network parameters that minimise the squared Frobenius norm given in (6.63).

$$L(\mathbf{W}_1, \mathbf{W}_2, \boldsymbol{s}_1, \boldsymbol{s}_2) = \left|\left|\mathbf{R}_1^C - \hat{\mathbf{R}}_1^C\right|\right|_2^2 + \left|\left|\mathbf{R}_2^C - \hat{\mathbf{R}}_2^C\right|\right|_2^2 \tag{6.63}$$

Experimental settings, datasets and results will be presented in the next sections.

6.3.2 Experimental Settings and Datasets

KG. We use the FB15k-237 dataset created by Toutanova et al. (2015) for training KGEs using the RelWalk model. This dataset has been introduced in link prediction in Section 6.2.3. FB15k-237 dataset contains 237 relation types for 14541 entities. To preserve the asymmetry property for relations, we consider that each relation $r^{<}$ in the relation set has its inverse $r^{>}$, so that for each triple $(h, r^{<}, t)$ in the KG $(t, r^{>}, h)$ is also in the KG. Thus as a total we have 474 relation types to be learnt (we refer to this extended version as FB15K-474). The train, test and validation parts of this dataset contains respectively $544, 230$, $40, 932$ and $35, 070$ tuples. Following the recommendations by the authors, RelWalk is trained on FB15K-474 using 100 mini-batches for 1000 epochs until convergence. The negative sampling rate was set to 50 and we learn KGEs of dimensionalities $d = 20, 50$ and 100. The matrix relational embeddings and entity embeddings produced by RelWalk are used in the subsequent experiments when learning supervised compositional operators.

Compositional constraints. To learn the proposed relation composition operator, we use the global-statistic constraints created by Takahashi et al. (2018) from FB15K-237 as follows. For a relation r, the authors define the *content set* $S(r)$ as the set of (h, t) pairs such that (h, r, t) is a fact in the KG. Likewise, they define $S(r_A \wedge r_B)$ as the set of (h, t) pairs such that $h \xrightarrow{r_A} e_1 \xrightarrow{r_B} t$ is a path in the KG. Next, $r_A \wedge r_B \rightarrow r_C$ is considered as a compositional constraint if their content sets are similar; that is, if $|S(r_A \wedge r_B) \cap S(r_C)| \geq 50$ and the Jaccard similarity between $S(r_A \wedge r_B)$ and $S(r_C)$ is greater than 0.4. They obtained

Table 6.5: Examples of compositional constraints $r_A \wedge r_B \rightarrow r_C$ along with Jacard scores, taken from Takahashi et al. (2018) dataset.

$r_A \wedge r_B$	r_C	Jacard score
`ceremony`$^{>}$ $\wedge$ `instance_of_recurring_event`$^{>}$	`category_of`$^{>}$	0.747
`adjustment_currency`$^{<}$ $\wedge$ `country`$^{<}$	`currency`$^{<}$	0.557
`sport`$^{<}$ $\wedge$ `team`$^{<}$	`athlete`$^{<}$	0.556
`nationality`$^{>}$ $\wedge$ `location_of_ceremony`$^{<}$	`type_of_union`$^{>}$	0.453

154 compositional constraints of the form $r_A \wedge r_B \rightarrow r_C$ after this filtering process. We abbreviate the name of this dataset to **RCC** from Relational Composition Constraint. Selected examples of compositional constrains are shown in Table 6.5.

Implementation details. We performed five folds cross-validation on the **RCC** dataset to train a supervised relation composition operator using our proposed method described in Section 6.3.1. Using a separate validation dataset, we set the initial learning rate for Adam to 5E-4 and minibatch size to 25. We apply dropout with rate of 0.5 and ℓ_2 regularisation with coefficient 1E-10 to avoid overfitting during training. For $d = 20$ dimensional embeddings, we use a single hidden layer of 300 neurones, whereas for $d = 50$ and 100 we used two hidden layers, where each has 600 neurones. In all settings, training converged after 25k epochs. The source code implementation of the proposed method, datasets and FB15K-474 KGEs are publicly available[8].

The section below describes evaluation tasks with experimental results.

6.3.3 Experimental Results

Recall that we assume that the composition of the two relations r_A and r_B is the relation r_C. We denote the pre-trained RelWalk embeddings for a relation r_x to be $\mathbf{R}_1^x$ and $\mathbf{R}_2^x$, where $x \in \{A, B, C\}$. The composed embedding for r_C is denoted by by $\hat{\mathbf{R}}_1^C$ and $\hat{\mathbf{R}}_2^C$. We evaluate the efficiency of the composed relation embedding in two tasks namely, relation composition ranking and triple classification as follows.

Relation Composition Ranking

This task aims to measure the similarity between a composed embedding for an unseen relation and all other relation embeddings. Following Takahashi et al. (2018), we rank the test relations r_L by its similarity to $\hat{r}_C$, the composed version of r_C, using the distance

[8] `https://github.com/Huda-Hakami/Relation-Composition-for-Knowledge-Graphs`

Table 6.6: Performance of relation composition ranking task.

	d=20			d=50			d=100		
Method	MR	MRR	Hits@10	MR	MRR	Hits@10	MR	MRR	Hits@10
Addition	238	0.010	0.012	250	0.008	0.019	247	0.007	0.000
Matrix Product	225	0.018	0.032	233	0.012	0.025	231	0.010	0.019
Hadamard Product	215	0.020	0.051	192	0.037	0.051	209	0.016	0.032
Supervised Relation Composition	**75**	**0.412**	**0.581**	**64**	**0.390**	**0.729**	**49**	**0.308**	**0.703**

function $d(r_L, \hat{r}_C)$ given by (6.64).

$$d(r_L, \hat{r}_C) = \left|\left|\mathbf{R}_1^L - \hat{\mathbf{R}}_1^C\right|\right|_F + \left|\left|\mathbf{R}_2^L - \hat{\mathbf{R}}_2^C\right|\right|_F \tag{6.64}$$

If the r_C is ranked higher than other test relations for $\hat{R}_C$, then it is considered better. We consider the 474 relation types in FB15K-474 as candidates (i.e., r_L) for this ranking process. Then, we use MR, MRR and Hits@10 to measure the performance of the composition.

Table 6.6 presents the average performance of relation compositions using five folds cross -validation on **RCC** compositional constraints. Lower MR, higher MRR and higher Hits@10 indicate better performance. As can be observed, the supervised relation composition achieves the best results for MR, MRR and Hits@10 with significant improvements over the unsupervised compositional operators. This observation is consistent for different dimensionality of relation embeddings $d = 20, 50, 100$. Hadmard product is the best among unsupervised relation compositional operators. However, the unsupervised operators collectively perform as the random baseline, which picks a relation type uniformly at random from the candidate relations.

Triple Classification

To evaluate the effectiveness of the learnt operators for generating composed relation embeddings, we consider the triple classification task using the composed relation embeddings and entity embedding. This task is presented in detail when we evaluated RelWalk embeddings in Section 6.2.3. Recall that this task aims to predict whether a triple (h, r, t) is a valid triple or not given entity and relation embeddings and a scoring function that maps the embeddings to a confidence score. We use the embeddings learnt by RelWalk for the entities and the relations in FB15k-474 and the joint probability $p(h, r, t)$ given by Theorem 2 to determine whether a relation r exists between two given entities h and t.

We perform five folds cross-validation on **RCC** compositional constraints. Once the proposed supervised relation composition is learnt using a training set, we perform triple classification for those triples in FB15K-474 testing set that are linked by the relation types in the held-out split of the compositional constraints. We evaluate the performance using the accuracy which is the percentage of the correctly classified test triples. We use the validation set to find a classification threshold T_{r_C} for each unseen relation r_c considering the predicted relation embeddings.

Table 6.7: Triple classification accuracy for the different relational compositional operators.

Method	d=20	d=50	d=100
Addition	68.9	70.44	69.45
Matrix Product	67.6	65.24	75.71
Hadamard Product	58.44	63.01	70.94
Supervised Relation Composition	**77.55**	**77.73**	**77.62**

The performance of the supervised and unsupervised relation composition operators for triple classification is shown in Table 6.7. Across the relational compositional operators and for different dimensionalities, the proposed supervised relational composition method achieves the best accuracy for this task. Despite increasing the dimensionality of relation embeddings from 20 to 100 leading to a complex model with a large number of parameters to be tuned using a small set of constraints as in **RCC**, the trained operator shows better performance in all the cases.

6.4 Summary

This chapter considered the problem of representing relations in KGs that include facts about the real world in the form of nodes linked by edges. We proposed RelWalk, a generative model of KGE and derived a theoretical relationship between the probability of a triple consisting of head, tail entities and the relation that exists between those two entities, and the embeddings of the corresponding entities and relations. In RelWalk, we represent entities by vectors and relations by matrices. We then proposed a learning objective based on the theoretical relationship we derived to learn entity and relation embeddings from a given knowledge graph. Experimental results on the link prediction and the triple classification tasks show that RelWalk performs similar to several previously proposed KGE learning methods. The key assumptions of RelWalk are validated by empirically analysing the relationship between such assumptions and the performance of the learnt embeddings from a KG. Moreover, we studied the compressibility of the learnt relation embeddings and discovered that using only 60% of the components, we can approximate the relation embeddings without any significant loss in performance.

This chapter also addressed the problem of representing novel relations by composing pre-trained relation embeddings in KGs. Given a set of compositional constraints over relations in the form $r_A \wedge r_B \rightarrow r_C$, we proposed a method that learns a supervised operator to map the relation embeddings of two relations to a new relation embedding. By doing so, it effectively tackled the problem of representing novel (or rare) relation types. Evaluating the predicted relation embeddings for triple classification task indicated the effectiveness of the proposed relation composition method. There are many further investigations and evaluations concering composition constraints and semantic composition models for novel relations that can be done.

We will now move to the last chapter to conclude this thesis by summarising our contributions, main findings and looking forward to future research.

7 Conclusion

This thesis was devoted to the task of learning semantic representations for relations between words, which is undoubtedly important in NLP applications. Earlier studies on representing word semantics from a large text corpus via deep learning techniques revealed the property of linguistic regularity in a space as linear translations between the embeddings of word-pairs related by a considered relation. In light of this characteristic about word embeddings, the focus of this thesis had been to investigate the compositional methods on pre-trained word embeddings to represent relations between words. Besides linguistic features in a text corpus to induce relational information, structured KGs provide us with real-world relational facts in the form of labelled edges between entities. The thesis was also concerned with embedding methodologies employed for representing relations in KGs.

This chapter concludes the work presented in this thesis in the following sections. Section 7.1 summarises the work presented in each chapter of the thesis. Then, Section 7.2 recaps the main contributions of the thesis under the research questions and issues raised in Chapter 1. In Section 7.3, some future directions that build upon the work conducted in this thesis are discussed.

7.1 Summary of Thesis

The thesis proposed multiple solutions to learn semantic representations for relations between words. There are mainly two approaches to capture relations between words, which are the pattern-based (i.e., requires co-occurring context between words) and the compositional (i.e., applies some operator on pre-trained word embeddings). Chapter 1 presented the main motivation for moving away from pattern-based to compositional approach for relations, as the former suffers from the sparsity problem because, even in a large corpus, not every related pair of words co-occur properly for their relation to be captured. Another line of research on relation representations is KG-based approaches that depend on an organised graph of relations (can be seen as connecting patterns in a text corpus) between entities covering real-world facts. The data sparsity problem also exists in KGs as a considerable number of facts are still missing despite the best efforts to create complete KGs because new entities are constantly emerging and new relations are formed between relations. Embedding entities and relations of a KG in a low-dimensional latent space show exciting success to expand otherwise sparse KGs. The research questions for which the study of this thesis focused were defined in Chapter 1 along with the contributions made to answer them.

In Chapter 2, a literature review concerning related topics was presented. First of all, the benefits of spending effort to learn relation semantics in the NLP field were discussed. Historical background about the two approaches to learn relations (pattern-based and compositional) and their limitations was presented. A discussion about the complementarity of the two sources of information for relations was also reviewed in the chapter. The last part of the chapter was about multi-relational KGs and learning relation representations in KGs using KGE methods.

A systematic study of unsupervised compositional operators for representing relations was presented in Chapter 3. For various word embedding models and evaluation tasks, PairDiff was the overall best. However, breaking down the performance by relation type revealed the fact that the best operator varied from one relation type to another. The chapter also presented a proposed supervised classification-based method that defines a discriminative feature space to measure relational similarities of word-pairs. The experimental results indicated that the effective features for measuring relational similarity are indeed ranked at the top by the proposed method.

Chapter 4 explored bilinear operators to represent a relation between two words using their pre-trained word embeddings. The chapter provided a mathematical analysis by computing the expected ℓ_2 loss that minimises the distance between relation representations from the bilinear operator of analogous word-pairs while maximises it for non-analogous pairs. The chapter showed that, under specified assumptions, the expected loss was independent of the bilinear terms.

Chapter 5 tackled the problem of learning parameterised compositional operators to represent relations between words. The chapter proposed two compositional operators modelled as non-linear neural networks. The first was called MnnPL, where the penultimate layer of a feed-forward neural network trained for classifying relation types of input word-pairs provided accurate representations that generalise well for out-of-domain relations. The second proposed operator, called CGRE, built on the first by incorporating relational patterns as a regulariser, and learnt in a self-supervised manner. Taken together, the successful performance reported by the proposed operators provided evidence that compositional methods can recover the relational regularities hidden inside word embedding spaces.

Finally, Chapter 6 looked at representing relations between entities in multi-relational

KGs. In particular, a proposed relational walk model for learning KGEs by performing a random walk over a KG was presented. The chapter also introduced a novel task – relation composition that predicts embeddings for novel relations from existing ones.

7.2 Main Contributions and Findings

This section provides a brief recap of the contributions along with the main findings in this thesis. The key research question considered in this thesis was follows:

"*Can we learn relation representations from word representations; and if so what are the appropriate methods and resources for achieving this?*"

To answer this main research question, a number of subsidiary questions had been raised, as presented in Section 1.3. These questions will be reviewed in this section in terms of the main findings of the research presented in the thesis.

1. *Given pre-trained word embeddings, what is the best unsupervised compositional operator to represent relations between words? and how appropriate is such an operator for various relation types?*
   It has been shown that prediction-based word embeddings encode features correlated with relational knowledge, which can be obtained via unsupervised PairDiff operator, but it remains unclear as to what is the best unsupervised operator to derive relation representations from word embeddings. **In Chapter 3 (Section 3.4), a systematic comparative study was conducted, which revealed the superiority of the PairDiff operator among various word embedding models and evaluation tasks**. The appropriateness of the PairDiff operator for different relation types was evaluated. Overall, our study showed that syntactic relations are easier to capture using the PairDiff operator than semantic relations. Besides, encyclopedic semantic relations have shown to be well organised under the PairDiff compared to Lexicographic semantic relations.

2. *Can we discover discriminating relational features from word representations to measure the relational similarity between two word-pairs?*
   As elaborated in Section 3.5, the attributional similarity between the corresponding arguments of two word-pairs can be seen as a reason for considering the two pairs as instances of the same relation. However, the features that accurately express the relational similarity between two word-pairs remain unknown. For this case, **a data-driven approach was proposed to discover representative space for relational similarity measurement from word representations.** It was found that the extracted features are efficient descriptors of semantic relations compared to linguistically-oriented methods.

3. *Can we systematically investigate a bilinear operator, which is parametrised by a 3D tensor, to map two given word embeddings into a vector representing a relation between the two words?*

   Despite the empirical success of the PairDiff operator for relations, it was remained unclear as to whether we can learn better parameterised operators to represent relations. **Thus, a theoretical analysis of generalised bilinear operators for relation representations that can be used to measure the distances between word-pairs was conducted in Chapter 4.** It was demonstrated that, if the word embeddings are standardised and uncorrelated, such an operator will be independent of bilinear terms, and can be simplified to a linear form, where the PairDiff is a special case. The general applicability of the theoretical result was demonstrated by empirically verifying underlying assumptions.

4. *Can we learn better compositional operators for relation representations from word embeddings?*

   To answer this question, neural network-based models were considered for the purpose of learning relation compositional operators. **Section 5.2 of Chapter 5 presented MnnPL, a compositional operator modelled as a non-linear neural network learnt in a supervised fashion.** It had been found that the proposed method could generalise to out-of-domain settings by representing word-pairs from unseen relations. This finding demonstrated the fact that simple supervised operators can accurately discover hidden relational features in word embeddings.

5. *Can we improve the performance of compositional relation representation methods by training such methods using the two sources of information namely: (a) word-embeddings of related pairs and, (b) co-occurring patterns extracted from a corpus?*

   As discussed in Section 2.6, bridging the gap between pattern-based and compositional approaches for learning relations can tackle the limitations of using each approach separately. To this end, In **Chapter 5 (Section 5.3), a context-guided relation embedding method was proposed that considered co-occurring patterns at training time, and only word embeddings for unseen word-pairs during the inference.** The proposed model was approached in a self-supervised manner to get rid of the need to annotate data. Regularising a compositional operator with relational patterns improved the performance and made the usage of compositional methods more efficient. Therefore, the combination of word representation features and relational pattern features are useful for learning relation representations.

6. *Given a KG, can we enrich the graph by inferring missing links using a theoretically motivated approach for relation and entity embeddings?*

   Despite the good empirical performance of heuristically defined KGE methods, theoretical understanding of KGEs is relatively underdeveloped. **This thesis attempted to fill this void by developing a theoretical model, relational walk, that performed a random walk over a KG and derived a scoring function that relates KGEs to the connections in the KG.** KGEs were learnt from a given KG such that the relationship given by the proven theorem was empirically satisfied. Accurate KGEs were leant from the derived objective for benchmark KGs, which in turn provided empirical evidence in support of the theory.

7. *Given pre-trained KGEs, can we infer embeddings for unseen (i.e., novel) relations using pre-trained embeddings for the existing relations?*
   Typically, KGE methods learn representations for entities and relations existing in a given KG, then such representations are used to detect unseen links between already seen entities and relations. Limitations arise, however, when previously unseen relation types are encountered during test time. **Chapter 6 (Section 6.3) tackled this problem by proposing a supervised relation composition operator to predict representations for novel relations.** The proposed operator efficiently outperformed its unsupervised counterparts on relevant evaluations.

After discussing each of subsidiary question, let us return to the main research question:

"*Can we learn relation representations from word representations; and if so what are the appropriate resources and methodologies for achieving this?*"

It can be stated that multiple resources assist in the learning of representations for relations between words. One of these sources were pre-trained word embeddings that attracted our attention because they: (a) succeeded initially in analogical reasoning, and (b) tackled the sparsity problem as exacted co-occurrences between two words are no longer required for their relation to be represented. The conducted work demonstrated that we can extract hidden information about relations from pre-trained word embeddings by learning compositional operators modelled by deep neural networks. According to the results of our study, it was also found that relational patterns (another source of relational knowledge) co-occurring between words in a corpus can be leveraged to boost the performance of relation embeddings when they were incorporated in the learning framework as a regularisation for a compositional operator. The thesis also took into consideration another ubiquitous source to learn relations, namely knowledge graphs in which facts are represented as nodes of entities connected through edges labelled by relation types. Such KGs have been used widely for various NLP related tasks. The concept of knowledge graph embeddings played a very important role in reasoning about relations between entities and thus enriching sparse knowledge graphs. In our work, the relationship between the connections in a graph and the embeddings of entities and relations were derived theoretically and used to learn efficient relation matrices that operate on entity vectors to score the relational triples.

Collectively, the analysis and results obtained in this thesis have shown that relation learning methods yield meaningful semantic representations for relations that would significantly benefit numerous NLP applications.

7.3 Future Work

Learning representations for semantic relations between words is a growing research area as understanding the underlying connections in a text span is an important requirement for intelligent systems. This section outlines possible future directions for learning relation representations. In particular, three major themes for future exploration are listed below.

Intrinsic Evaluation of Relation Embeddings

Ideally, we want to obtain relation embeddings that can perform efficiently on downstream tasks such as textual entailment and metaphor detection. However, evaluating relation embeddings on such tasks at development time is computationally expensive. Thus, during our work, we adopted relational similarity tasks as a proxy evaluation that correlated well with downstream tasks. Nonetheless, a direct interpretation of dimensions themselves, in a latent semantic space of relations, is still obscured.

In word representations, few studies have attempted to provide explanations of the dimensions learnt under prediction-based word embedding models. Tsvetkov et al. (2015), for example, proposed a qualitative intrinsic evaluation method called QVEC, which maps latent dimensions in word embeddings to linguistic-oriented features. Extending the QVEC measure from word-level representations to relation representations is possible, but not straightforward because the features that characterise relations are different from those associated with word meanings. While, for instance, the dimensions in word embeddings latently related to POS or senses such as *animal*, *food*, *motion*, relations can be characterised by contextual patterns such as *increase the risk* and *instance of*. Proposing an intrinsic evaluation of relation representations to facilitate qualitative analysis and interpretability is thus a potential research avenue.

Inference in Knowledge Graphs

Despite the efforts directed at developing KGs covering a wide range of information about entities and relations between them, it remains a challenging task to keep KGs up to date with the latest information because new entities are constantly emerging and new relations are formed between entities. Such missing knowledge can be populated by performing inference on a KG. Existing KGEs methods, including the one proposed in the thesis, RelWalk, reason about individual relations without considering connected paths of edges.

One way to extend inferences in a KG is to reason with a set o logical rules. For example, the rule $\mathsf{IsA}(X, Y) \wedge \mathsf{IsA}(Y, Z) \implies \mathsf{IsA}(X, Z)$ can be used to infer an IsA relation between X and Z if that relation is already satisfied between (X, Y) and (Y, Z). Specifying additional knowledge in the form of logical rules such as first-order Horn clauses is attractive because of its compactness. Consequently, prior proposals for injecting knowledge into KGs have used logical rules as the preferred knowledge representation method (Guo et al., 2016; Demeester et al., 2016; Ding et al., 2018; Minervini and Riedel, 2018). Numerous methods have been proposed for measuring the strength of a relation between two entities using embeddings learnt for relations and entities from a given KB. Given such a scoring formula, we can determine whether the head clause of a first-order Horn-style logical rule could be entailed by the body clause of the rule. However, it remains unclear as to what is the best method to compute the entailment score for a logical rule. A possible future work is to investigate the potential of learning a data-driven approach to find the best scoring formula satisfying the entailment constraints in a set of logical rules. These logical rules can be derived automatically via rule mining systems (Galárraga et al., 2015).

Relational Knowledge From Contextualised Word Embeddings

This thesis analysed and experimented semantics of relations considering static word embedding models. Each word in the vocabulary, at the end of training a model on co-occurrence statistics, is assigned a fixed context-free representation. From 2018 onwards, contextualised representations, obtained from deep neural language models, emerged as a method to generate dynamic representations of a word based on the surrounding contexts (Peters et al., 2018; Devlin et al., 2019). These contextual representation models have shown impressive performance when fine-tuned for many NLP tasks including question answering, entity recognition, etc. Interestingly, researchers also considered such rich semantic contextual representations for KG completion and relation learning (Baldini Soares et al., 2019; Yao et al., 2019; Petroni et al., 2019; Bouraoui et al., 2020). In particular, it has been shown that contextual language models can fill the blanks in relational questions such as "*London* is the capital of ?". As such, a worthwile future work is to employ contextualised representations for predicting lexical patterns of unobserved word-pairs to represent their relations.

Bibliography

Eneko Agirre, Oier López de Lacalle, and Aitor Soroa. 2014. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, 40(1):57–84.

Roee Aharoni and Yoav Goldberg. 2017. Towards string-to-tree neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 132–140, Vancouver, Canada. Association for Computational Linguistics.

Carl Allen and Timothy Hospedales. 2019. Analogies explained: Towards understanding word embeddings. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 223–231, Long Beach, California, USA. PMLR.

Mohammed Alsuhaibani and Danushka Bollegala. 2018. Joint learning of sense and word embeddings. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Mohammed Alsuhaibani, Danushka Bollegala, Takanori Maehara, and Ken-ichi Kawarabayashi. 2018. Jointly learning word embeddings using a corpus and a knowledge base. *PlOS ONE*, 13(3):e0193094.

Luis Espinosa Anke, Steven Schockaert, and Leo Wanner. 2019. Collocation classification with unsupervised relation vectors. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5765–5772.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2015. Rand-walk: A latent variable model approach to word embeddings. *arXiv preprint arXiv:1502.03520*.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. A latent variable model approach to PMI-based word embeddings. *Transactions of the Association for Computational Linguistics (TACL)*, 4:385–399.

Mohammed Attia, Suraj Maharjan, Younes Samih, Laura Kallmeyer, and Thamar Solorio. 2016. CogALex-v shared task: GHHH - detecting semantic relations via word embeddings.

In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex - V)*, pages 86–91, Osaka, Japan. The COLING 2016 Organizing Committee.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Elena Baralis, Luca Cagliero, Saima Jabeen, Alessandro Fiori, and Sajid Shah. 2013. Multi-document summarization based on the yago ontology. *Expert Systems with Applications*, 40(17):6976–6984.

John A Barnden and Mark G Lee. 2001. *Metaphor and Artificial Intelligence: A Special Double Issue of Metaphor and Symbol*. Psychology Press.

Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1183–1193, Cambridge, MA. Association for Computational Linguistics.

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

Matthew Berland and Eugene Charniak. 1999. Finding parts in very large corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 57–64, College Park, Maryland, USA. Association for Computational Linguistics.

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

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

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, Vancouver, Canada. Association for Computing Machinery.

Danushka Bollegala, Huda Hakami, Yuichi Yoshida, and Ken-ichi Kawarabayashi. 2019. Relwalk – a latent variable model approach to knowledge graph embedding.

Danushka Bollegala, Takanori Maehara, Yuichi Yoshida, and Ken-ichi Kawarabayashi. 2015. Learning word representations from relational graphs. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 2146–2152, Austin, Texas. AAAI Press.

Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2008. Www sits the sat: Measuring relational similarity on the web. In *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI)*, pages 333–337, Patras, Greece. IOS Press.

Danushka Bollegala, Yuichi Yoshida, and Ken-ichi Kawarabayashi. 2018. Using k-way co-occurrences for learning word embeddings. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 5037–5044, New Orleans, Louisiana USA.

Danushka Tarupathi Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2010. Relational duality: Unsupervised extraction of semantic relations between entities on the web. In *Proceedings of the 19th international conference on World Wide Web (WWW)*, pages 151–160, Raleigh, North Carolina, USA. ACM.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems (NeurIPS)*, pages 2787–2795, Red Hook, NY, USA. Curran Associates Inc.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, pages 301–306, San Francisco, California. AAAI Press.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Guillaume Bouchard, Sameer Singh, and Theo Trouillon. 2015. On approximate reasoning capabilities of low-rank vector spaces. In *2015 AAAI Spring Symposium Series*, pages 6–9, Stanford University, CA.

Zied Bouraoui, Jose Camacho-Collados, and Steven Schockaert. 2020. Inducing relational knowledge from BERT. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, New York, NY, USA.

Zied Bouraoui, Shoaib Jameel, and Steven Schockaert. 2018. Relation induction in word embeddings revisited. In *Proceedings of the 27th International Conference on Computa-*

tional Linguistics (COLING), pages 1627–1637, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

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

Michael J Cafarella, Michele Banko, and Oren Etzioni. 2006. Relational web search. In *Proceedings of the 15th International World Wide Web Conference (WWW)*, Edinburgh, UK.

Liwei Cai and William Yang Wang. 2018. KBGAN: Adversarial learning for knowledge graph embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1470–1480, New Orleans, Louisiana. Association for Computational Linguistics.

Jose Camacho-Collados, Luis Espinosa-Anke, Shoaib Jameel, and Steven Schockaert. 2019. A latent variable model for learning distributional relation vectors. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI*, pages 4911–4917, Macao, China. International Joint Conferences on Artificial Intelligence Organization.

Joseph B Casagrande and Kenneth L Hale. 1967. Semantic relationships in papago folk-definitions. *Studies in Southwestern ethnolinguistics*, pages 165–193.

Roger Chaffin and Douglas J Herrmann. 1984. The similarity and diversity of semantic relations. *Memory & Cognition*, 12(2):134–141.

Yin-Wen Chang and Chih-Jen Lin. 2008. Feature ranking using linear svm. In *Causation and Prediction Challenge*, pages 53–64.

Dawn Chen, Joshua C Peterson, and Thomas L Griffiths. 2017. Evaluating vector-space models of analogy. *The cognitive science society*, pages 1746–1751.

Wenye Chen, Huda Hakami, and Danushka Bollegala. 2019. Learning to compose relational embeddings in knowledge graphs. In *Proceedings of the 16th International Conference of the Pacific Association for Computational Linguistics (PACLING)*, Hanoi City, Vietnam.

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

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537.

D Alan Cruse, David Alan Cruse, and D A Cruse. 1986. *Lexical semantics*. Cambridge university press.

Rajarshi Das, Manzil Zaheer, Siva Reddy, and Andrew McCallum. 2017. Question answering on knowledge bases and text using universal schema and memory networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 358–365, Vancouver, Canada. Association for Computational Linguistics.

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

Thomas Demeester, Tim Rocktäschel, and Sebastian Riedel. 2016. Lifted rule injection for relation embeddings. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1389–1399, Austin, Texas. Association for Computational Linguistics.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 1811–1818, New Orleans, Louisiana USA.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Minneapolis, Minnesota. Association for Computational Linguistics.

Boyang Ding, Quan Wang, Bin Wang, and Li Guo. 2018. Improving knowledge graph embedding using simple constraints. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 110–121, Melbourne, Australia. Association for Computational Linguistics.

Pamela Downing. 1977. On the creation and use of english compound nouns. *Language*, 53(4):810–842.

Aleksandr Drozd, Anna Gladkova, and Satoshi Matsuoka. 2016. Word embeddings, analogies, and machine learning: Beyond king - man + woman = queen. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING)*, pages 3519–3530, Osaka, Japan. The COLING 2016 Organizing Committee.

Nguyen Tuan Duc, Danushka Bollegala, and Mitsuru Ishizuka. 2010. Using relational similarity between word pairs for latent relational search on the web. In *2010 IEEE/WIC/ACM*

International Conference on Web Intelligence and Intelligent Agent Technology, volume 1, pages 196–199. IEEE.

Nguyen Tuan Duc, Danushka Bollegala, and Mitsuru Ishizuka. 2011. Cross-language latent relational search: Mapping knowledge across languages. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, pages 1237–1242, San Francisco, California.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Luis Espinosa-Anke and Steven Schockaert. 2018. SeVeN: Augmenting word embeddings with unsupervised relation vectors. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, pages 2653–2665, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Kawin Ethayarajh. 2019. Rotate king to get queen: Word relationships as orthogonal transformations in embedding space. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3503–3508, Hong Kong, China. Association for Computational Linguistics.

Kawin Ethayarajh, David Duvenaud, and Graeme Hirst. 2019. Towards understanding linear word analogies. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3253–3262, Florence, Italy. Association for Computational Linguistics.

Miao Fan, Kai Cao, Yifan He, and Ralph Grishman. 2015. Jointly embedding relations and mentions for knowledge population. In *Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP)*, pages 186–191, Hissar, Bulgaria. INCOMA.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015a. 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, Denver, Colorado. Association for Computational Linguistics.

Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah A. Smith. 2015b. Sparse overcomplete word vector representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint*

Conference on Natural Language Processing (ACL-IJCNLP), pages 1491–1500, Beijing, China. Association for Computational Linguistics.

Stefano Federici, Simonetta Montemagni, and Vito Pirrelli. 1997. Inferring semantic similarity from distributional evidence: an analogy-based approach to word sense disambiguation. In *Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pages 90–97.

Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. 2008. 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.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: The concept revisited. *ACM Transactions on information systems*, 20(1):116–131.

John R Firth. 1957. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*.

Ronald A Fisher. 1915. Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population. *Biometrika*, 10(4):507–521.

Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning semantic hierarchies via word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1199–1209, Baltimore, Maryland. Association for Computational Linguistics.

Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2015. Learning semantic hierarchies: A continuous vector space approach. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):461–471.

Kata Gábor, Haïfa Zargayouna, Isabelle Tellier, Davide Buscaldi, and Thierry Charnois. 2017. Exploring vector spaces for semantic relations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1814–1823, Copenhagen, Denmark. Association for Computational Linguistics.

Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M Suchanek. 2015. Fast rule mining in ontological knowledge bases with AMIE. *The VLDB Journal*, 24(6):707–730.

Matt Gardner, Partha Pratim Talukdar, Bryan Kisiel, and Tom Mitchell. 2013. Improving learning and inference in a large knowledge-base using latent syntactic cues. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 833–838, Seattle, Washington, USA. Association for Computational Linguistics.

Dedre Gentner, Brian Bowdle, Phillip Wolff, Consuelo Boronat, et al. 2001. Metaphor is like analogy. *The analogical mind: Perspectives from cognitive science*, pages 199–253.

Lise Getoor and Ashwin Machanavajjhala. 2012. Entity resolution: theory, practice & open challenges. *Proceedings of the VLDB Endowment*, 5(12):2018–2019.

Roxana Girju, Adriana Badulescu, and Dan Moldovan. 2003. Learning semantic constraints for the automatic discovery of part-whole relations. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-HLT)*, pages 1–8. Association for Computational Linguistics.

Alex Gittens, Dimitris Achlioptas, and Michael W. Mahoney. 2017. Skip-gram â^' Zipf + uniform = vector additivity. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 69–76, Vancouver, Canada. Association for Computational Linguistics.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the NAACL Student Research Workshop*, pages 8–15, San Diego, California. Association for Computational Linguistics.

Arnold L Glass, Keith J Holyoak, and Nancy E Kossan. 1977. Children's ability to detect semantic contradictions. *Child Development*, pages 279–283.

Goran Glavaš and Simone Paolo Ponzetto. 2017. Dual tensor model for detecting asymmetric lexico-semantic relations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1757–1767, Copenhagen, Denmark. Association for Computational Linguistics.

Goran Glavaš and Ivan Vulić. 2018. Discriminating between lexico-semantic relations with the specialization tensor model. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 181–187, New Orleans, Louisiana. Association for Computational Linguistics.

Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. 2016. Jointly embedding knowledge graphs and logical rules. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 192–202, Austin, Texas. Association for Computational Linguistics.

Abhijeet Gupta, Gemma Boleda, and Sebastian Padó. 2017. Distributed prediction of relations for entities: The easy, the difficult, and the impossible. In *Proceedings of the 6th*

*Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 104–109, Vancouver, Canada. Association for Computational Linguistics.

Kelvin Guu, John Miller, and Percy Liang. 2015. Traversing knowledge graphs in vector space. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 318–327, Lisbon, Portugal. Association for Computational Linguistics.

Huda Hakami and Danushka Bollegala. 2017. Compositional approaches for representing relations between words: A comparative study. *Knowledge-Based Systems (KBS)*, 136:172–182.

Huda Hakami and Danushka Bollegala. 2019a. Context-guided self-supervised relation embeddings. In *Proceedings of the 16th International Conference of the Pacific Association for Computational Linguistics (PACLING)*, Hanoi City, Vietnam.

Huda Hakami and Danushka Bollegala. 2019b. Learning relation representations from word representations. In *Proceedings of the 1st Conference on Automated Knowledge Base Construction (AKBC)*, Amherst, MA.

Huda Hakami, Kohei Hayashi, and Danushka Bollegala. 2018. Why does PairDiff work? - a mathematical analysis of bilinear relational compositional operators for analogy detection. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, pages 2493–2504, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Huda Hakami, Angrosh Mandya, and Danushka Bollegala. 2017. Discovering representative space for relational similarity measurement. In *Proceedings of the 15th International Conference of the Pacific Association for Computational Linguistics (PACLING)*, pages 76–87, Yangon, Myanmar. Springer.

Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. OpenKE: An open toolkit for knowledge embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 139–144, Brussels, Belgium. Association for Computational Linguistics.

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

Kazuma Hashimoto, Pontus Stenetorp, Makoto Miwa, and Yoshimasa Tsuruoka. 2015. Task-oriented learning of word embeddings for semantic relation classification. In *Proceedings of the 19th Conference on Computational Natural Language Learning (CoNLL)*, pages 268–278, Beijing, China. Association for Computational Linguistics.

Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics (COLING)*, pages 539–545. Association for Computational Linguistics.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 94–99. Association for Computational Linguistics.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France. PMLR.

Shoaib Jameel, Zied Bouraoui, and Steven Schockaert. 2018. Unsupervised learning of distributional relation vectors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 23–33, Melbourne, Australia. Association for Computational Linguistics.

Yangfeng Ji and Jacob Eisenstein. 2013. Discriminative improvements to distributional sentence similarity. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 891–896, Seattle, Washington, USA. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Omer Levy, Daniel Weld, and Luke Zettlemoyer. 2019. pair2vec: Compositional word-pair embeddings for cross-sentence inference. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 3597–3608, Minneapolis, Minnesota. Association for Computational Linguistics.

David A Jurgens, Peter D Turney, Saif M Mohammad, and Keith J Holyoak. 2012. Semeval-2012 task 2: Measuring degrees of relational similarity. In *Proceedings of the 1st Joint*

Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, pages 356–364. Association for Computational Linguistics.

Christopher SG Khoo and Jin-Cheon Na. 2006. Semantic relations in information science. *Annual review of information science and technology*, 40(1):157–228.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 2015 International Conference on Learning Representations (ICLR)*, San Diego, CA, USA.

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text Generation from Knowledge Graphs with Graph Transformers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2284–2293, Minneapolis, Minnesota. Association for Computational Linguistics.

Natalia Konstantinova. 2014. Review of relation extraction methods: What is new out there? In *International Conference on Analysis of Images, Social Networks and Texts*, pages 15–28. Springer.

Maximilian Köper, Christian Scheible, and Sabine Schulte im Walde. 2015. Multilingual reliability and “semantic” structure of continuous word spaces. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 40–45, London, UK. Association for Computational Linguistics.

Timothee Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018. Canonical tensor decomposition for knowledge base completion. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pages 2863–2872, Stockholmsmässan, Stockholm Sweden. PMLR.

George Lakoff and Mark Johnson. 2008. *Metaphors we live by.* University of Chicago press.

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

Ni Lao and William W Cohen. 2010. Relational retrieval using a combination of path-constrained random walks. *Machine learning*, 81(1):53–67.

Ni Lao, Tom Mitchell, and William W. Cohen. 2011. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the 2011 Conference on Empirical*

Methods in Natural Language Processing (EMNLP), pages 529–539, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. 2008. Zero-data learning of new tasks. In *Proceedings of the 23rd National Conference on Artificial Intelligence*, pages 646–651. AAAI Press.

Daniel D Lee and H Sebastian Seung. 2001. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems (NIPS)*, pages 556–562.

Judith N Levi. 1978. *The syntax and semantics of complex nominals.* Academic Press New York.

Omer Levy and Yoav Goldberg. 2014. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the 18th Conference on Computational Natural Language Learning (CoNLL)*, pages 171–180, Ann Arbor, Michigan. Association for Computational Linguistics.

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

Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015b. 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, Denver, Colorado. Association for Computational Linguistics.

Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, and Xiaoyong Du. 2018. Analogical reasoning on Chinese morphological and semantic relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 138–143, Melbourne, Australia. Association for Computational Linguistics.

Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. 2015. Modeling relation paths for representation learning of knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 705–714, Lisbon, Portugal. Association for Computational Linguistics.

Tal Linzen. 2016. Issues in evaluating semantic spaces using word analogies. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 13–18, Berlin, Germany. Association for Computational Linguistics.

Shusen Liu, Peer-Timo Bremer, Jayaraman J Thiagarajan, Vivek Srikumar, Bei Wang, Yarden Livnat, and Valerio Pascucci. 2017. Visual exploration of semantic relationships in neural word embeddings. *IEEE transactions on visualization and computer graphics*, 24(1):553–562.

Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015. Topical word embeddings. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 2418–2424, Austin, Texas USA.

Robert Logan, Nelson F. Liu, Matthew E. Peters, Matt Gardner, and Sameer Singh. 2019. Barack's wife hillary: Using knowledge graphs for fact-aware language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5962–5971, Florence, Italy. Association for Computational Linguistics.

Federico López, Benjamin Heinzerling, and Michael Strube. 2019. Fine-grained entity typing in hyperbolic space. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 169–180, Florence, Italy. Association for Computational Linguistics.

Kevin Lund and Curt Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior research methods, instruments, & computers*, 28(2):203–208.

Lianbo Ma, Peng Sun, Zhiwei Lin, and Hui Wang. 2019. Composing knowledge graph embeddings via word embeddings. *arXiv preprint arXiv:1909.03794*.

Pranava Swaroop Madhyastha, Xavier Carreras, and Ariadna Quattoni. 2014. Learning task-specific bilexical embeddings. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, pages 161–171, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Pranava Swaroop Madhyastha, Xavier Carreras, and Ariadna Quattoni. 2015. Tailoring word embeddings for bilexical predictions: An experimental comparison. In *3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, USA.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*. Cambridge university press.

Eliabeth Marshman. 2002. The cause-effect relation in a biopharmaceutical corpus: English knowledge patterns. In *Terminology and knowledge engineering*, pages 89–94.

Diana McCarthy. 2007. Word sense disambiguation: Algorithms and applications. *Computational Linguistics*, 33(2):255–258.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations (ICLR)*, Scottsdale, Arizona, USA.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems (NIPS)*, pages 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. 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, Atlanta, Georgia. Association for Computational Linguistics.

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

George A Miller and Walter G Charles. 1991. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28.

Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 777–782, Atlanta, Georgia.

Pasquale Minervini, Thomas Demeester, Tim Rocktäschel, and Sebastian Riedel. 2017. Adversarial sets for regularising neural link predictors. In *Proceedings of the 33th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, Sydney, Australia.

Pasquale Minervini and Sebastian Riedel. 2018. Adversarially regularising neural NLI models to integrate logical background knowledge. In *Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL)*, pages 65–74, Brussels, Belgium. Association for Computational Linguistics.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 1003–1011. Association for Computational Linguistics.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244, Columbus, Ohio. Association for Computational Linguistics.

Jeff Mitchell and Mirella Lapata. 2009. Language models based on semantic composition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 430–439, Singapore. Association for Computational Linguistics.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–1429.

Dunja Mladenić, Janez Brank, Marko Grobelnik, and Natasa Milic-Frayling. 2004. Feature selection using linear classifier weights: interaction with classification models. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 234–241.

Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. OpenDialKG: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 845–854, Florence, Italy. Association for Computational Linguistics.

Jiaqi Mu and Pramod Viswanath. 2018. All-but-the-top: Simple and effective post-processing for word representations. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, Vancouver, Canada.

Brian Murphy, Partha Talukdar, and Tom Mitchell. 2012. Learning effective and interpretable semantic models using non-negative sparse embedding. In *Proceedings of the 2012 International Conference on Computational Linguistics (COLING)*, pages 1933–1950, Mumbai, India. The COLING 2012 Organizing Committee.

Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. 2012. Patty: a taxonomy of relational patterns with semantic types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1135–1145. Association for Computational Linguistics.

Preslav Nakov. 2008. Improved statistical machine translation using monolingual paraphrases. In *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI)*, volume 178, page 338, Patras, Greece. IOS Press.

Vivi Nastase, Preslav Nakov, Diarmuid O Seaghdha, and Stan Szpakowicz. 2013. Semantic relations between nominals. *Synthesis lectures on human language technologies*, 6(1):1–119.

Vivi Nastase and Stan Szpakowicz. 2003. Exploring noun-modifier semantic relations. In *Fifth international workshop on computational semantics (IWCS-5)*, pages 285–301.

Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. 2019. Learning attention-based embeddings for relation prediction in knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4710–4723, Florence, Italy. Association for Computational Linguistics.

Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. 2015. Compositional vector space models for knowledge base completion. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 156–166, Beijing, China. Association for Computational Linguistics.

Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 2018. A novel embedding model for knowledge base completion based on convolutional neural network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 327–333, New Orleans, Louisiana. Association for Computational Linguistics.

Dat Quoc Nguyen. 2017. An overview of embedding models of entities and relationships for knowledge base completion. *arXiv preprint arXiv:1703.08098*.

Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, and Mark Johnson. 2016. STransE: a novel embedding model of entities and relationships in knowledge bases. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 460–466, San Diego, California. Association for Computational Linguistics.

Maximilian Nickel, Xueyan Jiang, and Volker Tresp. 2014. Reducing the rank in relational factorization models by including observable patterns. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1179–1187.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2015. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic embeddings of knowledge graphs. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 1955–1961, Phoenix, Arizona USA.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, volume 11, pages 809–816, Bellevue, WA, USA.

Pinal Patel, Disha Davey, Vishal Panchal, and Parth Pathak. 2018. Annotation of a large clinical entity corpus. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2033–2042, Brussels, Belgium. Association for Computational Linguistics.

Heiko Paulheim. 2017. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web*, 8(3):489–508.

Derek C Penn, Keith J Holyoak, and Daniel J Povinelli. 2008. Darwin's mistake: Explaining the discontinuity between human and nonhuman minds. *Behavioral and Brain Sciences*, 31(2):109–130.

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

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. 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, New Orleans, Louisiana. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Jay Pujara, Eriq Augustine, and Lise Getoor. 2017. Sparsity and noise: Where knowledge graph embeddings fall short. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1751–1756, Copenhagen, Denmark. Association for Computational Linguistics.

James Pustejovsky, Joseé Castano, Jason Zhang, Maciej Kotecki, and Brent Cochran. 2001. Robust relational parsing over biomedical literature: Extracting inhibit relation. In *Biocomputing 2002*, pages 362–373. World Scientific.

Ning Qian. 1999. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151.

Xiang Ren, Zeqiu Wu, Wenqi He, Meng Qu, Clare R Voss, Heng Ji, Tarek F Abdelzaher, and Jiawei Han. 2017. Cotype: Joint extraction of typed entities and relations with knowledge bases. In *Proceedings of the 26th International Conference on World Wide Web (WWW)*, pages 1015–1024. International World Wide Web Conferences Steering Committee.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD)*, pages 148–163. Springer.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 74–84, Atlanta, Georgia. Association for Computational Linguistics.

Anna Rogers, Aleksandr Drozd, and Bofang Li. 2017. The (too many) problems of analogical reasoning with word vectors. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 135–148, Vancouver, Canada. Association for Computational Linguistics.

Stephen Roller, Douwe Kiela, and Maximilian Nickel. 2018. Hearst patterns revisited: Automatic hypernym detection from large text corpora. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 358–363. Association for Computational Linguistics.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic. Association for Computational Linguistics.

Gaetano Rossiello, Alfio Gliozzo, Robert Farrell, Nicolas Fauceglia, and Michael Glass. 2019. Learning relational representations by analogy using hierarchical Siamese networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 3235–3245, Minneapolis, Minnesota. Association for Computational Linguistics.

F de Saussure. 1959. Course in general linguistics. *New York: Philosophical Library.*

Natalie Schluter. 2018. The word analogy testing caveat. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 242–246, New Orleans, Louisiana. Association for Computational Linguistics.

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

Karl-Michael Schneider. 2005. Weighted average pointwise mutual information for feature selection in text categorization. In *European Conference on Principles of Data Mining and Knowledge Discovery (PKDD)*, pages 252–263, Porto, Portugal. Springer.

Steven Schockaert and Henri Prade. 2014. Completing symbolic rule bases using betweenness and analogical proportion. In *Computational Approaches to Analogical Reasoning: Current Trends*, pages 195–215. Springer.

Vered Shwartz and Ido Dagan. 2016. Path-based vs. distributional information in recognizing lexical semantic relations. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex - V)*, pages 24–29, Osaka, Japan. The COLING 2016 Organizing Committee.

Vered Shwartz and Ido Dagan. 2018. Paraphrase to explicate: Revealing implicit noun-compound relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1200–1211, Melbourne, Australia. Association for Computational Linguistics.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2389–2398, Berlin, Germany. Association for Computational Linguistics.

Rion Snow, Daniel Jurafsky, and Andrew Y Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In *Advances in neural information processing systems (NIPS)*, pages 1297–1304.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013a. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems (NIPS)*, pages 926–934.

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

Yu Su, Honglei Liu, Semih Yavuz, Izzeddin Gür, Huan Sun, and Xifeng Yan. 2018. Global relation embedding for relation extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 820–830, New Orleans, Louisiana. Association for Computational Linguistics.

Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Shengping Liu. 2019. Leverage lexical knowledge for Chinese named entity recognition via collaborative graph network. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3830–3840, Hong Kong, China. Association for Computational Linguistics.

Alona Sydorova, Nina Poerner, and Benjamin Roth. 2019. Interpretable question answering on knowledge bases and text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4943–4951, Florence, Italy. Association for Computational Linguistics.

Ryo Takahashi, Ran Tian, and Kentaro Inui. 2018. Interpretable and compositional relation learning by joint training with an autoencoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2148–2159, Melbourne, Australia. Association for Computational Linguistics.

Lucien Tesnière. 1959. *Eléments de syntaxe structurale*. Klincksieck.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1499–1509, Lisbon, Portugal. Association for Computational Linguistics.

Stephen Tratz and Eduard Hovy. 2010. A taxonomy, dataset, and classifier for automatic noun compound interpretation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 678–687, Uppsala, Sweden. Association for Computational Linguistics.

Gautami Tripathi and S Naganna. 2015. Feature selection and classification approach for sentiment analysis. *Machine Learning and Applications: An International Journal*, 2(2):1–16.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33rd*

International Conference on Machine Learning (ICML), pages 2071–2080, New York, NY, USA.

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

Peter Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 680–690, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Peter D. Turney. 2005. Measuring semantic similarity by latent relational analysis. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1136–1141, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Peter D Turney. 2006. Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416.

Peter D Turney. 2008. The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research*, 33:615–655.

Peter D Turney. 2012. Domain and function: A dual-space model of semantic relations and compositions. *Journal of Artificial Intelligence Research*, 44:533–585.

Peter D Turney and Michael L Littman. 2005. Corpus-based learning of analogies and semantic relations. *Machine Learning*, 60(1-3):251–278.

Peter D Turney, Michael L Littman, Jeffrey Bigham, and Victor Shnayder. 2003. Combining independent modules to solve multiple-choice synonym and analogy problems. *arXiv preprint cs/0309035*.

Peter D Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188.

Tu Vu and Vered Shwartz. 2018. Integrating multiplicative features into supervised distributional methods for lexical entailment. In *Proceedings of the 7th Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 160–166, New Orleans, Louisiana. Association for Computational Linguistics.

Ekaterina Vylomova, Laura Rimell, Trevor Cohn, and Timothy Baldwin. 2016. 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, Berlin, Germany. Association for Computational Linguistics.

Chengyu Wang, Xiaofeng He, and Aoying Zhou. 2019a. Improving hypernymy prediction via taxonomy enhanced adversarial learning. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pages 7128–7135, Hawaii, USA.

Chengyu Wang, Xiaofeng He, and Aoying Zhou. 2019b. SphereRE: Distinguishing lexical relations with hyperspherical relation embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1727–1737, Florence, Italy. Association for Computational Linguistics.

Ruijie Wang, Yuchen Yan, Jialu Wang, Yuting Jia, Ye Zhang, Weinan Zhang, and Xinbing Wang. 2018. Acekg: A large-scale knowledge graph for academic data mining. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1487–1490. ACM.

Koki Washio and Tsuneaki Kato. 2018a. Filling missing paths: Modeling co-occurrences of word pairs and dependency paths for recognizing lexical semantic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1123–1133, New Orleans, Louisiana. Association for Computational Linguistics.

Koki Washio and Tsuneaki Kato. 2018b. Neural latent relational analysis to capture lexical semantic relations in a vector space. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 594–600, Brussels, Belgium. Association for Computational Linguistics.

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

Han Xiao, Minlie Huang, and Xiaoyan Zhu. 2016. TransG : A generative model for knowledge graph embedding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2316–2325, Berlin, Germany. Association for Computational Linguistics.

Yang Xu, Gareth JF Jones, JinTao Li, Bin Wang, and ChunMing Sun. 2007. A study on mutual information-based feature selection for text categorization. *Journal of Computational Information Systems*, 3(3):1007–1012.

Josuke Yamane, Tomoya Takatani, Hitoshi Yamada, Makoto Miwa, and Yutaka Sasaki. 2016. Distributional hypernym generation by jointly learning clusters and projections. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING)*, pages 1871–1879, Osaka, Japan. The COLING 2016 Organizing Committee.

Bishan Yang, Wen-tau Yih, Xiadong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the 2015 International Conference on Learning Representations (ICLR)*, Vancouver, Canada.

Roman Yangarber, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. 2000. Automatic acquisition of domain knowledge for information extraction. In *Proceedings of the 18th conference on Computational linguistics (COLING)*, pages 940–946. Association for Computational Linguistics.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. KG-BERT: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.

Wenpeng Yin and Hinrich Schütze. 2016. Learning word meta-embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1351–1360, Berlin, Germany. Association for Computational Linguistics.

Dani Yogatama, Manaal Faruqui, Chris Dyer, and Noah A. Smith. 2015. Learning word representations with hierarchical sparse coding. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 87–96, Lille, France. JMLR.org.

Hee-Geun Yoon, Hyun-Je Song, Seong-Bae Park, and Se-Young Park. 2016. A translation-based knowledge graph embedding preserving logical property of relations. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 907–916, San Diego, California. Association for Computational Linguistics.

Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, and Minlie Huang. 2018. Augmenting end-to-end dialogue systems with commonsense knowledge. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 4970–4977, New Orleans, Louisiana USA.

Omnia Zayed, John Philip McCrae, and Paul Buitelaar. 2018. Phrase-level metaphor identification using distributed representations of word meaning. In *Proceedings of the Workshop on Figurative Language Processing*, pages 81–90, New Orleans, Louisiana. Association for Computational Linguistics.

Wen Zhang, Jiawei Hu, Yang Feng, and Qun Liu. 2018. Refining source representations with relation networks for neural machine translation. In *Proceedings of the 27th International*

Conference on Computational Linguistics (COLING), pages 1292–1303, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yuanzhe Zhang, Kang Liu, Shizhu He, Guoliang Ji, Zhanyi Liu, Hua Wu, and Jun Zhao. 2016. Question answering over knowledge base with neural attention combining global knowledge information. *arXiv preprint arXiv:1606.00979*.

Alisa Zhila, Wen-tau Yih, Christopher Meek, Geoffrey Zweig, and Tomas Mikolov. 2013. Combining heterogeneous models for measuring relational similarity. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1000–1009, Atlanta, Georgia. Association for Computational Linguistics.