Computation on Meanings:
Content-based Feature Analysis for Semantic Interpretation

DAT TAN HUYNH

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

in

Information Sciences and Engineering

University of Canberra

February, 2015

University of Canberra

Abstract

Computation on Meanings: Content-based Feature Analysis for Semantic Interpretation

DAT TAN HUYNH

Building a computer system that has extraordinary capacity to understand human language has become one of the greatest challenges. One of the barriers in building such a system is to construct a semantic interpretation of human language.

Many prior approaches, which have been proposed to address the issue, can be categorized into knowledge-based approaches and content-based approaches. While knowledge-based approaches utilise human-crafted knowledge repositories to construct semantic interpretation, content-based approaches analyse a large amount of unstructured text data available. Although knowledge-based approaches produce outstanding performance compared to content-based approaches on semantic interpretation as well as semantic distance,they are limited themselves within in particular text knowledge domains. On the other hand, despite content-based approaches have the ability to process unstructured text data on different domains and languages, there are certain aspects that are worthy of considerations.

First, most of the prior content-based approaches popularly use a single type of

feature aspects to construct word meaning representation. This raises a concern how multiple feature aspects can be used to model meaning representation. Secondly, experiments of the content-based approaches using various sets of features were undertaken, their performances tested on the task of measuring semantic distance are still under expectation.

The main focus of this research is to propose new content-based approaches for semantic interpretation of human language. By undertaking semantic analysis on large amount of unstructured text available, our proposed methods have presented new sets of features for semantic interpretation. On the one hand, multiple aspect sets of features have been proposed, which help to cover semantic meanings of words in different angles. On the other hands, feature transformations and combinations have been proposed to not only reduce size of feature dimensions, but also encorage the interaction between different aspects of word features.

The contributions of this research are listed as the followings:

– Relational Feature Analysis For Semantic Interpretation : this approach constructs representing features of a word by considering its relations extracted from the word's local contexts as well as the hidden aspects built from the sets of relations. The effectiveness of generated features is evaluated based on the task of measuring semantic distance. Experimental results have demonstrated the promising capacity of the relation-based features in modelling word mean-

ing compared to traditional context-based features when tested on the same benchmarks.

– Conceptual Topic Analysis For Semantic Distance: this approach introduces a new way to construct a semantic profile of word meanings and measuring semantic distance by using topical clues from surrounding contexts to characterise meanings of a word. With the experiment on various standard benchmarks, the method demonstrates outstanding performance compared to related methods using topical information.

– Multi-way Feature Analysis for Semantic Interpretation: this approach proposes a tensor-based technique for semantic interpretation by building meaning representation of words directly from text and does not require pre-existing linguistic knowledge. Taking in to account structural information such as word order and syntactic information, the method that utilises tensor analysis to build representation of word meaning. This content-based model demonstrates significantly improved performance when compared to a robust baseline model on a number of semantic distance measures.

The success of semantic interpretation of words contributes for building a reliable metric for semantic distance, which involves in most tasks of natural language processing and understanding.

ACKNOWLEDGMENTS

I am grateful to many people who have assisted me throughout my research journey.

First, I am thankful for the generous support provided by the chair of supervision panel, Associate Professor Dat Tran. I am honoured to have an opportunity to work with him and thankful for his endless guidance, support, and encouragement.

I would also like to express my sincere gratitude to supervisor Assistant Professor Wanli Ma, for his supports and feedbacks, collaborations during my PhD journey. I am grateful to Professor Dharmendra Sharma, Mr Hanh Huynh, Dr Kim Le for consistent helps and supports within the Faculty of Information Science and Engineering and The Faculty of ESTEM.

I specially acknowledge the University of Canberra and family of the late W. J. Weeden for their postgraduate research scholarship to conduct this research. I also thank the Turing Centre at the University of Washington, Stanford NLP Group for providing me tools, data, and resource to conduct experiments during my research, and special thank Wikipedia for such a valuable data

repository. I also thank the Institude for Applied Ecology for providing access to the high end cluster during my experiments.

I would like to thank everyone in the Faculty of ISE, ESTEM–especially colleagues in CIKADA group including Dr Tuan Hoang, Phuoc Nguyen, Khoa Nguyen, Tan Vo, Dung Pham, Dinh Phung–for providing helpful assistances, discussions and productive collaborations. I also specially thank Harriet Searcy for taking the hard part in proofreading my thesis.

Finally, I specially thank my wife and my families back home in Vietnam for all their love and support, for letting me to follow my dreams.

DEDICATION

To my wife, and my parents

TABLE OF CONTENTS

Page

List of Figures xvii

List of Tables xix

Glossary xxi

Chapter 1: Introduction 1
1.1 Meaning in Context 2
1.2 Representing Meaning 2
1.3 Problem Statement 3
1.4 Proposing Approach 10
1.5 Thesis Contribution 12
1.6 Thesis Outline 12

Chapter 2: Theoretical Background 15
2.1 Introduction 15
2.2 Semantic Representation 16
2.2.1 Feature Generation 16
2.2.2 Document-based Features 17
2.2.3 Word-based Features 19
2.2.4 Pattern-based features 20
2.2.5 Concept-based Features 20
Explicit Concept-based Features 21
Latent Concept-based Features 22
2.2.6 Learning-based Features 23
2.3 Semantic Distance 24

2.3.1 Evaluating Semantic Distance . . . 25
2.4 Approaches to Semantic Distance . . . 27
2.4.1 Knowledge-based Approaches . . . 27
2.4.2 Content-based Approaches . . . 35
2.5 Conclusion . . . 41

Chapter 3: Relational Feature Analysis For Semantic Interpretation . . . 43
3.1 Introduction . . . 43
3.2 Meaning Representation Using Relations in Context . . . 46
3.2.1 Relation Extraction Algorithm . . . 47
3.2.2 Confidence Function for Extracted Triples . . . 49
3.2.3 Learning Parameters for the Confidence Function . . . 51
3.2.4 The relational Semantic Space of Words . . . 53
3.3 Meaning Representation Using N-Gram Context . . . 54
3.3.1 Building the Word Semantic Space Using NGram Context . . . 55
3.3.2 Weighting Filters . . . 56
3.4 Inducing Hidden Concepts from Semantic Space . . . 57
3.5 Experiment . . . 59
3.5.1 Testing Benchmarks . . . 59
3.5.2 Corpus . . . 59
3.5.3 NGram Word Features Extraction . . . 60
3.5.4 Relational Feature Extraction . . . 61
3.5.5 Relational Feature Extraction with Confidence . . . 62
3.5.6 Hidden Feature Extraction . . . 63
3.5.7 Semantic Distance . . . 64
3.6 Evaluation . . . 65
3.6.1 Overall results compared to other content-based methods . . . 65
3.6.2 Feature Generation Analysis . . . 67
3.7 Conclusion . . . 70

Chapter 4: Conceptual Topic Analysis for Semantic Distance . . . 73
4.1 Introduction . . . 73
4.2 Topical Context Analysis . . . 75

4.2.1 Explicit Topic Analysis . . . 76
4.2.2 Extracting Seeds For Wikipedia Concepts . . . 76
4.2.3 Latent Topic Analysis . . . 80
4.3 A Combination Features for Word Meaning Representation . . . 82
4.3.1 Context-based Features and Topical Seeds . . . 82
4.3.2 Context-based Features and Topical Features . . . 84
4.3.3 Evaluation of Feature Combinations . . . 87
4.4 Experiments . . . 88
4.4.1 Testing Benchmarks . . . 88
4.4.2 Text Corpus . . . 89
4.4.3 Extracting latent topics and seeding representatives . . . 89
4.4.4 Extracting explicit topics and seeding representatives . . . 90
4.4.5 Choosing Parameters for Feature Combination . . . 91
4.5 Evaluation . . . 92
4.5.1 Overall results compared to other topic-based methods . . . 92
4.5.2 Latent Topic Features vs. Explicit Topic Features . . . 95
4.5.3 Linear Combination vs. Seed-based Combination . . . 96
4.6 Conclusion . . . 96

Chapter 5: Multi-way Feature Analysis For Semantic Interpretation . . . 97
5.1 Introduction . . . 98
5.2 Related Work . . . 100
5.2.1 Word Meanings in Distributional Contexts . . . 100
5.2.2 Creating the Word Semantic Space . . . 102
5.2.3 Tensor Analysis . . . 103
5.3 Tensor Analysis for Meaning Representation . . . 104
5.3.1 Multi-way Arrays . . . 104
5.3.2 Modelling the Semantic Space using a Three-way Tensor . . . 106
Memory Tensor . . . 107
Capturing the Closeness of Triples . . . 112
Computing on Word Meaning . . . 114
A Measure of Syntagmatic Association . . . 115
A Measure of Paradigmatic Association . . . 116

A Measure of Semantic Distance . . . 117
5.3.3 Multi-way Latent Feature Analysis . . . 118
Parallel Factor Analysis . . . 119
Representing Word Meaning . . . 120
Measuring Semantic Distance . . . 121
5.4 Experiment . . . 122
5.4.1 Text Repository . . . 123
5.4.2 Extracting Relations as Triples . . . 123
5.4.3 Encoding Triples into Three-way Tensor . . . 124
5.4.4 Measuring Semantic Distance based on Syntagmatic and Paradigmatic Strengths . . . 125
5.4.5 Multi-way Latent Feature Analysis . . . 126
5.5 Evaluation and Discussion . . . 127
5.5.1 Overall Results . . . 127
5.5.2 Syntagmatic Strength vs Paradigmatic Strength . . . 129
5.5.3 Further remarks . . . 131
5.6 Conclusion . . . 132

Chapter 6: Conclusion . . . 135
6.1 Summaries of the Proposed Approaches . . . 135
6.2 Overall Results from Measuring Semantic Distance . . . 138
6.3 Addressing Research Questions . . . 139
6.3.1 What strategies can be used to utilise relations between words for semantic interpretation? . . . 139
6.3.2 What are the effects of topical analysis for word meaning representation? . . . 140
6.3.3 What are the strategies to use words in contexts and structural information on the task of semantic interpretation? . . . 141
6.3.4 What strategies can be used to improve the performance of content-based methods on the task of measuring semantic distance? . . . 142
6.4 Future Investigations . . . 142
6.5 Future Applications . . . 143
6.6 Conclusion . . . 146

Bibliography 147

LIST OF FIGURES

Figure Number — Page

4.1 Details of the explicit topic × seed matrix $M_{exp}[topic, seed]$. Each row of the matrix is a vector of seeds, where only K-best components are selected while the rests are set to zeros 79

4.2 Details of the latent topic × seed matrix $M_{lat}[topic, seed]$. Each row of the matrix is a representation of a latent topics, which is constituted by a vector of seeds. Only the values of K best seeds are selected, and the rest are set to zeros. 81

5.1 The cube represents a model of a third-order (three-way) tensor: $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$. I, J, and K are the number of dimensions for each mode (way) (Kolda & Bader 2009) 104

5.2 Slices of a third-order tensor. Horizontal slices, lateral, frontal slices are obtained by fixing the first, second, and third model of the tensor respectively. 105

5.3 Fibers of a third-order tensor. Column, Raw, or Fiber columns are obtained by varying either the first, second, or the third mode of the tensor respectively and fixing the rest remaining modes (Kolda & Bader 2009) 106

5.4 A three-way tensor as a result of constructing three triples T_1=(*garlic*, *kill*, *bacteria*), T_2=(*garlic*, *kill*, *fungi*), T_3=(*garlic*, *kill*, *virus*). The respective slice for each mode is also depicted when fixing the respective mode, which shows how the encoding values are assigned 113

5.5 Graphical representation of PARAFAC. A tensor $\mathcal{X}$ is approximated into a combination of three loadings matrices: $A_{I \times Z}$, $B_{J \times Z}$, and $C_{K \times Z}$ 119

5.6 Procedures to conduct an experiment for the first Tensor model . . . 122

5.7 Procedures to conduct an experiment for the second Tensor model . 125

5.8 Correlation results (%) on each datasets have been presented when changing the trade-off parameter α 130

LIST OF TABLES

Table Number Page

2.1 A brief comparison features used for semantic representation 18

3.1 List of features used to identify whether a triple is reliable or unreliable. These features was inherited from the work of Fader et al. (2011) in Open Information Extraction. 50

3.2 Experiment using MTurk for turning parameters. The best Spearman's correlation score (%) was obtained with $FF = 2$, $IVF = 1$ on NGram word features, and with $r = 0.7$ on the confident threshold. Results of the related works on the same dataset were also presented. 60

3.3 List of features using their weightings updated from the Logistic Regression algorithm on training examples 62

3.4 The experimental results (%) with different content-based methods on WS-353 datasets using Spearman's rank correlation (ρ). In the first row, results from the related work are presented. The results of our proposed features are showed in two last rows of the table. 65

3.5 Experimental results (%) from various content-based methods on RG-65 datasets using Spearman's rank correlation (ρ). The first section shows the results from the related works. The results of our proposed features are also demonstrated in two last sections of the table 66

3.6 Correlation results (%) for NGram word features and relational word features when applying a frequency filter (FF=2) and information value filter (IVF) with the semantic distance tested on WS-353 dataset using Spearman's rank correlation (ρ). The best value is bolded and its corresponding parameter value is underlined 68

3.7 Correlation results (%) from relational features when applying different thresholds r on the confidence value for each triple with the semantic distance tested on the WS-353 dataset using Spearman's rank correlation (ρ). The best value bolded and its corresponding parameter value is underlined. 69

4.1 Experiment on MTruk for tuning parameters. Four different feature combinations that require parameter turning were experimented. For each feature combination, we selected values of parameters that returned the best correlation results during the semantic distance measurements 88

4.2 Comparison results with different content-based methods on WS-353 datasets using Spearman's rank correlation (ρ). † indicates the results using the selected parameters from the independent dataset 93

4.3 Comparison results with different content-based methods on RG-65 datasets using Spearman's rank correlation (ρ). 95

5.1 An example of indexed and initialized terms from the sentence "*Garlic can also kill harmful bacteria, fungi, and viruses*" 107

5.2 Experiment on MTurk for turning the α parameter. The best Spearman's correlation score was obtained with $\alpha = 0.6$ 126

5.3 Correlation results for different tensor-based features used on the semantic distance measure, and tested on the WS-353 dataset using Spearman's rank correlation (ρ). Symbol ‡ indicates value of the α parameter selected from a independent test. Symbol $\star$ indicates the best results over a range of selected parameters. The result from the LSA baseline method is also presented. 128

5.4 Correlation results for different tensor-based features used on the semantic distance measure, and tested on the RG-65 dataset using Spearman's rank correlation (ρ). Symbol ‡ indicates value of the α parameter selected from a independent test. Symbol $\star$ indicates the best results over a range of selected parameters. The result from the LSA baseline method is also presented. 129

6.1 A summary of the best results of our proposed features tested on the WS-353 dataset for the task of semantic distance measurement. . . . 138

6.2 A summary of the best results of our proposed features tested on the RG-65 dataset for the task of semantic distance measurement. 139

GLOSSARY

KB: Knowledge-based

CB: Content-based

FF: Frequency Filter

IVF: Information Value Filter

LSA: Latent Semantic Analysis

HAL: Hyperspace Analogue to Language

ESA: Explicit Semantic Analysis

VSM: Vector Space Model

NMF: Non-negative Matrix Factorization

SVD: Singular Vector Decomposition

SSA: Salient Semantic Analysis

LDA: Latent Dirichlet Allocation

DF: Document Frequency

ESA: Explicit Semantic Analysis

IDF: Inverse Document Frequency

IG: Information Gain

MI: Mutual Information

PMI: Pointwise Mutual Information

TFIDF: A term weighting approach based on Term Frequency and Inverse Document Frequency

WWW: World Wide Web

WS-353: Word Similarity dataset with 353 pairs of words.

RG-65: A dataset created by (Rubenstein & Goodenough 1965) containing 65 pairs of words.

MTURK: A dataset created by (Radinsky et al. 2011) containing 287 pairs of words.

PARAFAC: Parallel Factor Analysis

Chapter 1
INTRODUCTION

The recent explosion of World Wide Web (WWW) and the expansion of Web social interactions over time has created an enormous amount of inexpensive human language text, contributing to the importance of intelligent access and utilisation to this data. At the same time, building a computer system that has the capacity to understand human language has become one of the greatest challenges. One of the barriers in building such a system is to construct unstructured human language into concepts, or something the system can understand and process. This task is referred to as the *semantic interpretation* of human language.

This chapter provides the reasons for conducting the present research on semantic interpretation. First, we overview the mechanism by which humans understand meanings in context and how the meaning representation can be used for computational processing. Second, we identify various research problems and argue why this research should be conducted. Finally, we overview our contributions by proposing methodologies to address the research issues.

1.1 Meaning in Context

Humans have an extraordinary capacity to infer meanings of words even when they have never met the words before. To figure out what a word actually refers to in a human language, people utilise their long-term knowledge to reason meanings for the word. Despite a word being unknown, its surrounding contexts can contribute to a reasonable guess. Consider the example sentences (1)–(3) (Huynh et al. 2014):

(1) *Everyone thinks pho is delicious.*
(2) *Ingredients of pho include broth, rice noodles, herbs, and meat.*
(3) *Pho is served in most Vietnamese restaurants.*

In the example sentences, the word "*pho*" is not seen in English dictionaries. It is a foreign term and its meaning is not known in English. Nevertheless, even if people have never heard of the word, they can still make a reasonable assumption about its meanings based on its accompanying text. This is because elements that appear in the contexts of "*pho*" such as "*restaurant*", "*broth*", "*ingredients*", and "*rice noodles*" allow humans to assume that its meanings are related to a kind of meals served in a restaurant.

1.2 Representing Meaning

Unlike people, who are capable of utilising their knowledge to generalize word meanings, the expectation of a computer's capacity to understand word semantics is more complicated. This is because a computer does not possess prior knowledge for its

reasoning. Its meaning reference tasks need to start from scratch. Secondly, with the enormous amount of text data currently available on the WWW, regardless of domains, languages and cultures, the issue has been raised to a new level of challenge.

To enable a computer to be able to manipulate meanings in context, the computer needs to model text information into concepts or something that the system can understand (Gabrilovich & Markovitch 2007). This task is intended to create a *semantic representation* of a text. In other words, *semantic representation* is the task of capturing the meaning contexts, generalising them into higher abstract levels where contextual clues are disclosed, so that they can be compared to, and differentiated from, each other automatically. In the previous example, the identified surrounding contexts of "*pho*" contain keywords such as "*ingredients*", "*broth*", "*rice*", "*noodles*", "*herbs*", and "*meat*", which are key identifiers of hidden topics such as "*food*". Compared, and contrasted, with other words from the same hidden topic, meanings of a specific word can be disclosed.

1.3 Problem Statement

With the explosion of textual data on the Web, the task of semantic representation has attracted significant work in the field. Starting from fundamental ideas embodies in the *distribution hypothesis* that words are used and occur in the same contexts tends to purport similar meanings (Harris 1954), many works have been proposed which aim to construct the semantic representation of word meanings.

By categorising features used for meaning representation, various works have been grouped into different catetories. *Word-based feature* approaches simply utilise words in text contents as features of representation (Harris 1954, Turney 2001, Lin 1998*b*, Agirre et al. 2009). In *Pattern-based feature* approaches, features of the presentations are designed by sets of rules and patterns (Turney 2006, Agirre et al. 2009). *Concept-based feature* approaches consider conceptual information from text as the key aspects to modelling features of the representation (Gabrilovich & Markovitch 2007, Hassan & Mihalcea 2011, Radinsky et al. 2011). *Latent feature* approaches aims to induce latent information from texts for meaning representation (Deerwester et al. 1990, Dinu & Lapata 2010). In *learning-based feature* approaches, features of representation are induced directly from texts (Bengio et al. 2006, Mnih & Hinton 2007, Collobert & Weston 2008, Huang et al. 2012). Although these kinds of features have successfully delivered the meaning representation of words[1], certain aspects require more consideration.

Firstly, from the *local context* perspective, the issue of modelling word meanings can be addressed by using its accompanying text with in a limited section of texts such as a sentence. This can be seen from the previous example containing the unknown word "*pho*". Various approaches have been presented in the line with this idea. In particular, word-based feature approaches and pattern-based feature approaches are typical initiatives, which directly use words in contexts as the represented features to

[1]Refer to Section 2.2 for comprehensive analysis of these works

address the issue. However, in the works using word-based features, any words can be accepted as represented features as long as they appear in a sliding window context. In this sense, *there is a lack of well-designed feature selection which selects reliable words in contexts for feature presentation.*

Additionally, pattern-based feature approaches utilise patterns, in which a pair of words appears together in context, as represented features for word meaning. For instance, the patterns "*the X used the Y to*" and "*the X shaped the Y in*" can be regarded as the pattern-based features of the word pair "*carpenter:wood*" (Turney 2006). These kinds of patterns only reflect the degree of relatedness between words rather than the degree of word similarity reflected by hyponymy (i.e. "*dog:cat*"), tropynymy (i.e. "*nibble:gorge*"), and antonymy (i.e. "*short:long*") relationships. Thus, *there are certain limitations on prior pattern-based features for meaning representation that consider only relatedness relationships between words for modelling word meanings.* To address this issue of feature representation in local contexts, answers to the research question: **What strategies can be used to utilise relations between words for semantic interpretation?** need to be investigated.

Secondly, from the *global context* perspective, meanings of a word are influenced by topical information from its larger areas such as documents, categories. Dumais (2004) proposed Latent Semantic Analysis (LSA), which considers word document co-occurrences to induce word meaning representation using latent topic features. Similarly, based on the relationships between Wikipedia documents and their con-

tent words, Gabrilovich & Markovitch (2007) proposed Explicit Semantic Analysis (ESA) to create meaning representation of words using Wikipedia concepts as explicit features. Such methods are typical works that only focus on relationships between words and documents to induce topic feature representation in global contexts, *but ignores the the possibility that topical information from local contexts could contribute to determining word meanings as well as meaning representation.* Moreover, most prior work on semantic interpretation has, up to this stage, focused only on single aspect features such as *word-based features* (Harris 1954, Turney 2001, Lin 1998*b*, Agirre et al. 2009), *topic-based features* (Gabrilovich & Markovitch 2007, Hassan & Mihalcea 2011, Radinsky et al. 2011, Deerwester et al. 1990, Dinu & Lapata 2010), with each type of feature contributing to particular aspects of meaning representation. *This raises the prospect that the combination of word-based features and topical features could significantly contribute to modelling word meaning representation.* To address this concern, findings from our second research question: **What strategies can be used to model word meanings by using topical information?** need to be investigated.

Thirdly, in the task of semantic interpretation, word meanings can be modelled by considering their distributions within contexts. In this case, word meanings are represented by collecting word co-location frequencies and encoding them into a high-dimensional *context vector* using the Vector Space Model (VSM), which assumes the existence of a word semantic space (Turney et al. 2010). However, one of the limita-

tions of this representation is that the context vector does not incorporate structural information from a text such as word order or syntactical aspects (Turney et al. 2010, Symonds et al. 2012). Moreover, in the task of understanding word meaning, roles of syntax and word order are significant. According to De Saussure (1916/1996), two fundamental word associations are believed to disclose word meaning: *syntagmatic* and *paradigmatic* associations. While a syntagmatic association refers to the relations between words that have strongly syntactical dependencies, a paradigmatic association between words expresses that they are able to exchange without alternating the original meanings of the entire context. These associations are heavily influenced by the way human reason word meanings and use word order and syntactic information as the basic structure for meaning manipulation. This emphasizes the importance of structural information in processing word meanings as well as in semantic processing in general. However, context vector representation ignores this kind of structured information. *This raises the possibility that an alternative way to represent word meanings can be found which also utilises the structural information.* To address this concern, findings from our third research question: **What are the strategies to use words in contexts and structural information on the task of semantic interpretation?** need to be investigated.

By considering the ways various approaches handle feeding information to build word semantic representation, recent works on semantic analysis can be classified

into two families: *Knowledge-based* (KB) methods and *Content-based* (CB) methods. On the one hand, the KB methods operate mainly using existing knowledge repositories such as WordNet (Miller et al. 1990), thesauri, and language dictionaries, which are manually constructed through human expertise[2] (Leacock & Chodorow 1998, Wu & Palmer 1994, Resnik 1995, Jiang & Conrath 1997, Resnik 1999, Budanitsky & Hirst 2006, Agirre et al. 2009, Hirst & St-Onge 1998). Others utilise the large amounts of human-crafted world knowledge such as is seen in Wikipedia (Yeh et al. 2009, Gabrilovich & Markovitch 2007, Strube & Ponzetto 2006, Hassan & Mihalcea 2011, Radinsky et al. 2011, Zesch et al. 2008). The KB algorithms tend to be straightforward, and often produce promising results for semantic analysis based on the reliability of human knowledge repositories. However, using pre-defined knowledge repositories results in certain limitations in KB methods. First, the amount of human crafted data is limited. It requires high concentration and human expertise to create such rich knowledge repositories. Secondly, such repositories are limited in particular text domains and languages, and are unable to process the variety of text domains available, especially when processing Web texts.

To overcome the limitations of KB methods, CB methods utilise the knowledge information from the unstructured texts available. They aim to transform the representation of word meanings in free text into a structure that can be made accountable. By doing so, CB methods are able to operate on different text domains and in differ-

[2]Refer to Section 2.4 for detailed analysis of these methods

ent languages. Typical works that have been conducted by analysing text contents range from methods using document global contexts (Dumais 2004, Dinu & Lapata 2010) to word local contexts (Turney 2001, Lin 1998*b*, Huang et al. 2012, Agirre et al. 2009). However, the performance of these CB methods is far below that of KB methods when tested and compared with human judgement on semantic distance measurement. *This raises a major concern about whether there is a way to improve the performance of the CB method using standard testing benchmarks.* Therefore, findings from our fourth research question: **What strategies can be used to improve the performance of content-based methods on the task of measuring semantic distance?** need to be investigated.

In summary, we have presented the major concerns about CB methods and issues in term of feature representation. We have also raised the following research questions for investigation:

1. What strategies can be used to utilise relations between words for semantic interpretation?

2. What are the effects of topical analysis for word meaning representation?

3. What are the strategies for using words in contexts and structural information on the task of semantic interpretation?

4. What strategies can be used to improve the performance of content-based meth-

ods on the task of measuring semantic distance?

1.4 Proposing Approach

Finding answers to the above research questions is the main objective of this research. As each of the questions concerns particular aspects of feature representation, our target is to conduct different content-based approaches, each of which will address particular aspects of meaning representation.

In responding to the first question, we have proposed a new approach for meaning representation. The method focuses on the impact of relations extracted from the text to construct the semantic representation of words. In this approach, we utilise lexical syntactic information as well as statistical quantities to access the quality of each extraction, which is then used to construct the meaning representation of words. This approach addresses two important issues related to feature representation in local context. First, the method introduce the ability to select reliable word features for meaning representation rather than accepting features as any words that appear in a sliding window. Secondly, it encodes unstructured text into triples, which facilitates the creation meaning representation that use both relatedness and similarity associations between words. Details of this work are presented in Chapter 3 and parts of Chapter 5.

To address the second question, we propose a new method, which examines local contexts to induce the latent topics of words. The topics will be used as feature

representation for word meanings. Latent topics have been induced in two different ways: (1) using explicit concepts appearing in local contexts, and (2) using word co-location in unstructured text data. The method also examines different feature combinations which integrate word-based features and latent topic features for meaning representation. The proposed approach addresses two important issues. First, it targets modelling feature representation using topical information from local contexts. Second, it demonstrates the capacity to combine word features and topic features, as well as their performance in this combination. Details of this work are presented in Chapter 4.

To address the third question, multiple aspects of word features have been used for semantic interpretation. We propose another new method which encodes relations between words in contexts and structural information into a a three-way tensor. The tensor model facilitates representation of word meanings as well as measurement of their semantic distance. The proposed approach addresses two important issues. First, it shows how an alternative representation of word meanings using a three-way tensor can be used instead of a context vector. Secondly, the tensor structure also enables to the direct use of relatedness and similarity associations in meaning representation along with measuring of semantic distance. Details of this approach are presented in Chapter 5.

To address the last question, all of the proposed approaches are evaluated by measuring semantic distance on standard benchmarks. The results are used as concrete

evidence, which address the performance concerns of content-based methods.

1.5 Thesis Contribution

This thesis embodies several contributions:

1. We propose a new method to model meaning representation using local context information. Represented features are created by considering the lexical syntactics and statistical quantities of reliable relations between words.

2. We formulate a new feature combination approach which integrates words in contexts and their latent topics for semantic representation.

3. We propose a new approach to semantic representation using multiple aspect features. The method encodes words in contexts and structural information into a three-way tensor which facilitates the representation of word meanings as well as the measurement of semantic distance.

4. The proposed method for feature generations present significantly improved results in semantic distance measurement compared to popular methods on the same benchmarks.

1.6 Thesis Outline

The outline of the thesis is as follows: after the introduction in Chapter 1, we briefly discuss background theories as well as related work in Chapter 2. While Chapter 3

presents feature generation using relations in local context, Chapter 4 examines topic-based analysis for meaning representation. In Chapter 5, we present an approach for semantic interpretation using multiple aspect features. Finally, the conclusion to the thesis is presented in Chapter 6.

Chapter 2
THEORETICAL BACKGROUND

This chapter introduces the background theories to meaning representation in contexts as well as that for semantic distance measurement. We first focus on discussing the methodologies that generate feature representation from various aspects. Secondly, we provide an overview of related work on measuring semantic distance. Information presented in this chapter provides the basic understandings used in the chapters that follows.

2.1 Introduction

One of the biggest barriers in advancing the power of computers is that they understand very little about meaning in human language. With the improvement in web searching, especially the expansion of social media, the task of understanding human language given the large amount of data has become more challenging. Semantic analysis is an important task towards understanding natural language. A goal of such analysis is to represent pieces of text as concepts, or something that computer systems can understand. This task is called *semantic interpretation* (Gabrilovich & Markovitch 2007).

While representing textual information is one part of semantic analysis, measuring

semantic distance between texts becomes another important evaluation task. Given the complexity of human language, textual information can be expressed in various ways but with the same or related meaning. The task of *semantic distance* is to measure the similarity/relatedness between the meaning of texts (Rubenstein & Goodenough 1965).

In the remaining parts of this chapter, we mainly focus on analysing typical works that explore *semantic interpretation* and *semantic distance*. Section 2.2 discusses feature representation in various aspects such as *document-based* features, *word-based* features, *concept-based* features,and *learning-based* features. An overview of semantic distance is then presented in Section 2.3, and typical approaches to it are reviewed in Section 2.4 under two different categories: the *knowledge-based* approaches and the *content-based* approaches.

2.2 Semantic Representation

2.2.1 Feature Generation

The Vector space model (VSM) (Salton et al. 1975) has been successful in modelling text information. VSM models textual information as a collection of dimensional features, which reflect meanings from different angles. For instance, each word is represented as a point in high-dimensional space, where the dimensions represent contextual features. Various feature aspects have been attempted in recent research including *document-based* features (Dumais 2004, Gabrilovich & Markovitch 2007),

word-based features (Lin 1998*b*, Turney 2006, Agirre et al. 2009), *concept-based* features which includes *explicit concept* (Gabrilovich & Markovitch 2007, 2009, Hassan & Mihalcea 2011) and *latent (hidden) concept* (Dumais 2004, Dinu & Lapata 2010), and *learning-based* features which are induced by a learning process (Bengio et al. 2006, Mnih & Hinton 2007, Collobert et al. 2011). Each feature aspect comprises important characteristics in different angles and usages, which are highlight in the Table 2.1. Detailed analysis on these feature aspects will be presented in the following sections.

2.2.2 Document-based Features

One of fundamental ideas in using *document-based* features for semantic representation is that words that appear in the same documents share common meanings in the document contexts. Dumais (2004) presented semantic representation of words by considering relations between words appearing in the same documents. Features were also generated by smoothing a word-document co-occurrence matrix using Singular Vector Decomposition (SVD) (De Lathauwer et al. 1994). Gabrilovich & Markovitch (2007) proposed document-based features for word semantic representation based on the Wikipedia repository. This method utilises word document co-occurrences within the entire Wikipedia corpus. Representation of a word was designed as a vector of Wikipedia documents, where the weighting of each respective feature was measured as the important level of the word within a document using TF×IDF algorithm (Salton

Table 2.1: A brief comparison features used for semantic representation

Feature Aspects	Characteristics	Strength	Weakness
Document-based Features	- Features generated from relations between words and documents - A word-document co-occurrence matrix used for estimate feature weightings	- simple implementation - adaptable to domains and languages - work well on large documents	- insufficient in short texts - limitted capacity to disclose semantic aspects in texts
Word-based Features	- Utilise relationships between words in local contexts. - A word-word co-occurrence matrix for feature weightings	- Explore semantics in local context - Intergrate structure information such as syntactic and word orders	- High dimensional vector
Pattern-based Features	- Feature generated using designed rules and patterns	- Straightforward approaches - Rules and patterns designed manually and depending on applications	- Required language expertises - Language dependency
Concept-based Features	- Utilise and explore pre-defined concepts in texts - Features generated using relationship between words and concepts	- Benefits from enriched knowledge repositories - Straightforward approaches - Adapt to use in different domains	- Require large amount of pre-defined knowledge - Language dependency
Learning-based Features	- Features induced from large amount of unstructured text - Feature weightings depending on the input data	- Provide large coverage and consistency for different domains	- Complex learning process - Difficult to integrate language expertise into the model

et al. 1975) .

2.2.3 Word-based Features

In most natural language processing tasks, local context information plays an important role in extracting the feature representation of text. Word features extracted from surrounding contexts have been popularly used to represent word meanings. In the early time, Harris (1954) proposed a distributional structure to construct representation of word meanings using high frequency co-occurring words in context. Similarly, Turney (2001) presented semantic representation of a word using words that were highly associated with each other. The authors used the Point-wise Mutual Information (PMI) as a weighting mechanism. Lin (1998*b*) extended the distributional hypothesis by presenting syntactical pattern-based features for word representation. Given a focus word, its semantic representation was constructed using surrounding words that had syntactical relations with the focus word. According to the authors, the higher the frequency words attached to each other via a syntactical relationship, the more important that word feature would be. Agirre et al. (2009) present a "bag-of-words" approach to semantic similarity between words. Each word in a dataset was represented by surrounding words that were associated with the focus word in a local context (a sliding window).

2.2.4 Pattern-based features

Designing language-based patterns to capture textual information has been popularly used in natural language processing (NLP), especially in information extraction (Etzioni et al. 2008). One of the benefits for rule-based work is that it is simple to implement and operate on particular domains. In semantic analysis, pattern-based features are also used particularly based on the characteristic of word orders, and lexical and syntactic information. Turney (2006) extends the ideas of the distributional structure of words to word pairs. The representation of a pair of words is constructed using a VSM model where each feature is a lexical pattern that the pair is frequently associated with. For example, the pair *carpenter:wood* is represented by pattern-based features such as "*the X used the Y to*" and "*the X shaped the Y in*" as both the pair and the patterns are highly associated. The authors utilised a syntactic dependency parser to extract the syntactic relationships between a word and its governing words. For each word in a text corpus, its syntactic dependency template was extracted as a feature for the word's representation. For instance, the pattern "*cooks* ⟨word⟩ *delicious*" could be a feature for the nouns such as "*food*", "*meals*", and "*pasta*".

2.2.5 Concept-based Features

The intuition behind using concept-based features for modelling textual information is that humans do not judge meanings of text at the word level but on the higher

abstract level such as that of concepts, categories, or topics where the meanings can be organized (Gabrilovich & Markovitch 2007). Most concept-based feature generation for meaning representation are grouped into two families: *explicit concept-based* features and *latent concept-based* features.

Explicit Concept-based Features

Gabrilovich & Markovitch (2007) proposed Explicit Semantic Analysis (ESA) to represent textual information by a vector of Wikipedia concepts (explicit concepts). The method is operated by using particular structures of Wikipedia articles such as Wikipedia links and concepts, as well as the large amount of text information available. Given a word that appears in a Wikipedia text, its semantic representation was a weighted vector of Wikipedia concepts where the weighting was measured by the co-occurrence association between the word and each particular concept. This work was later extended by Radinsky et al. (2011) who integrated the semantic representation using concepts from different domains such as Flickr image tags, and Del.icio.us bookmarks.

In a similar line of work that uses Wikipedia concepts as features for word representation, Hassan & Mihalcea (2011) proposed Salient Semantic Analysis (SSA) that utilised "*salient concepts*" from Wikipedia to enrich feature representation as well was the use of weighting schema. According to the authors, a "*salient concept*" is a Wikipedia concept that has not been labelled with a Wikipedia link. Instead of

relying on the Wikipedia link structure, the authors matched the Wikipedia concepts to the Wikipedia text to discover "*salient concepts*". Using PMI as the fundamental weighting schema as well as a specific weighting filter, the authors presented word semantics through a vector of Wikipedia concepts.

Latent Concept-based Features

While explicit concepts can be extracted using particular knowledge-based repositories, latent concepts can be induced from scratch using large amounts of text. In this case, feature representation of a text was constructed to describe word meanings beyond individual words. Some works have attempted to use latent features for semantic representation. Deerwester et al. (1990) and Dumais (2004) proposed Latent Semantic Analysis (LSA) to model word and document representation as vectors of latent features. These authors tried to address the problem of matching words in queries to words in documents. Different words can be expressed in different ways in writing, but will probably refer to the same or related conceptual topics. The approach aimed to overcome the deficiencies of term-matching retrieval by treating the unreliability of term-document association data as a statistical problem. The authors assumed that "there is some underlying latent semantic structure in the text data that is partially obscured by the randomness of word choice with respect to retrieval"(Deerwester et al. 1990). They used statistical techniques, such as Singular Vector Decomposition (SVD), to estimate this latent structure. A representation of

terms and documents were described as a vector of latent semantic features.

Dinu & Lapata (2010) have proposed a probabilistic framework to represent word meanings in context. They model the meanings of isolated words as a probability distribution over a set of latent senses. Assume that a target word w_i found in a corpus shares a global set of meanings or senses $Z = \{z_k|k : 1 \dots K\}$, the meanings of each individual target word w_i was described as a distribution over this set of senses. More formally, a target w_i was represented by the following vector:

$$v(w_i) = \langle\ P(z_1|w_i), ..., P(z_K|w_i)\ \rangle \tag{2.1}$$

where component $P(z_1|w_i)$ was the probability of sense z_1 given target word w_i, component $P(z_2|w_i)$ was the probability of sense z_2 given w_i and so on. The senses $z_1 \dots z_K$ were latent and could be seen as a means of reducing the dimensionality of the original co-occurrence matrix. The authors used either non-negative matrix factorization (NMF) or latent Dirichlet allocation (LDA) to construct the latent structure, which then encoded word meanings into a vector representation of latent features.

2.2.6 Learning-based Features

Methods to learn features are motivated by the idea that the semantic representation of words can be generated by the interactions of different factors on different levels such as word level, document level, or topic level. Deep learning adds the assumption

that these factors are organized into multiple levels, corresponding to different levels of abstraction or composition. Varying numbers of layers and layer sizes can be used to provide different amounts of abstraction. Neural Language Models (Bengio et al. 2006, Mnih & Hinton 2007, Collobert & Weston 2008) have been shown to be very powerful at language modelling, a task where models are asked to accurately predict the next word given previously seen words. The models learns simultaneously a distributed representation of each word along with the probability function for word sequences, expressed in terms of these representations. Similarly, Collobert et al. (2011) proposed a single multiple layer learning system that automatically learns word representation almost from scratch. Instead of training from a given set of training examples, the method uses vast amount of unlabelled training data. The system was designed flexibly to be able learn the representation for multiple NLP tasks. Huang et al. (2012) integrated learning features learned from local contexts (a sliding window) to global document contexts for the task of word semantic representation, which was then evaluated via the task of semantic similarity.

2.3 Semantic Distance

Semantic distance is a measurement of degree of relatedness/similarity between two units of language in terms of their meanings (Turney 2006). The units of language may be words, phrases, sentences, paragraphs, or documents. For instance, the two nouns "*traffic*" and "*street*" have more related meanings than the two nouns "*traffic*" and

"*garden*" and are thus semantically closer. Moreover, humans measure the semantic distance between words based not only on the text surface itself but also on their senses. For instance, the word "*bank*" has a meaning semantically close to "*money*" in a situation it is linked to the concept of "*monetary*", however, its meaning is also related to the word "*fish*" in the sense of a "*water body*".

Semantic distance can be expressed in two ways: "*semantic similarity*" and "*semantic relatedness*". In general, semantic similarity is a subset of semantic relatedness. In many cases, they are inter-changeable. Two concepts are considered to be semantically similar if there is a hyponymy, antonymy, or troponymy relation between them. Two concepts are considered to be semantically related if there is any lexical semantic relation between them. In other words, similar concepts tend to share a number of common properties. For instance, "*Pho*" and "*noodle soup*" are semantically similar as they are both hyponyms of "*food*", made from "*broth*", "*meat*" and "*vegetables*", and served in *restaurants*. On the other hand, semantically related concepts may not have any common properties but they have lexical relations between them. For instance, "*Pho*" and "*meat*" are semantically related as "*Pho*" includes "*meat*".

2.3.1 Evaluating Semantic Distance

Humans are adept at estimating semantic distance, however, different people judge the distance differently. When proposing a method for estimating semantic distance,

researchers need to have specific ways to evaluate the reliability of a method. Early work in the task of semantic distance has proposed various metrics to evaluate semantic distance. As these have been used over time and thus verified, they are considered as the standard benchmarks for the task of evaluation semantic distance.

Firstly, Rubenstein & Goodenough (1965) proposed a quality benchmark to measure the semantic similarity between words. They conducted quantitative experiments with human subjects (51 in all) who were asked to rate 65 English word pairs on a scale from 0.0 to 4.0, the greater the similarity of meanings, the higher the number as their semantic distance. The word pairs chosen ranged from being almost synonymous to completely unrelated. However, they were all noun pairs and which were semantically close and also semantically similar; the dataset did not contain word pairs that are semantically related but not semantically similar. The subjects repeated the annotation after two weeks and the new distance values had a Pearsons correlation r of 0.85 with the old ones. Additionally, Miller & Charles (1991) also conducted a similar study on 30 word pairs taken from the Rubenstein-Goodenough's pairs. These annotations had a high correlation (r = 0.97) with the mean annotations of Rubenstein & Goodenough (1965).

Word similarity test collection (WS-353) was designed to evaluate the semantic similarity measurement (Finkelstein et al. 2001). The collection is divided into two sets. The first set contains 153 word pairs along with their similarity scores assigned by 13 subjects. The second set contains 200 word pairs, with their similarity assessed

by 16 subjects. All the subjects experimented in both sets are near-native command of English. Their instructions were to estimate the relatedness of words in pairs on a scale from 0 (totally unrelated words) to 10 (very much related or identical words). In practice, the combination set that contains 353 words along with their mean similarity scores was most often used for testing purposes.

Finally, the MTurk dataset contains 287 pairs of words (Radinsky et al. 2011). Unlike WS-353, a computer was used to automatically draw up word pairs from words whose frequently occur together in large text domains. The relatedness of these pairs of words was then evaluated using human annotators, as done in the WS-353 dataset.

2.4 Approaches to Semantic Distance

Measuring the semantic distance between words is an important problem in lexical semantics. It has applications in many natural language processing tasks, such as Textual Entailment, Word Sense Disambiguation or Information Extraction, and other related areas like Information Retrieval. Previous work in the field of semantic representation and semantic distance can be categorised into two categories such as *knowledge-based (KB) approaches* and *content-based (CB) approaches*. This division is based on the way the methods handle the feeding information.

2.4.1 Knowledge-based Approaches

To a certain extent, the knowledge-based (KB) approaches refer to bodies of work that utilise knowledge repositories by any means such as crowd-sourcing or expertise. They

contrast to those that use unstructured text available without any human assistance. KB approaches have the advantage of being generally applicable to new texts and domains, without the needs for costly and perhaps error-prone parsing or semantic analysis. They require only human-crafted KB repositories.

One popular KB repository is WordNet (Miller et al. 1990), which is considered as a lexical database for the English. WordNet groups nouns, verbs, adjectives and adverbs into sets of synonyms called synsets, which provide short definitions and usage examples, and record a number of relations among these synonym sets or their members. WordNet can thus be seen as a combination of dictionary and a thesaurus. There are several typical work that have used WordNet as a language resource over the years (Leacock & Chodorow 1998, Wu & Palmer 1994, Resnik 1995, Jiang & Conrath 1997, Resnik 1999, Budanitsky & Hirst 2006, Agirre et al. 2009, Hirst & St-Onge 1998).

One of the simple approaches to measure semantic distance between words involves computing the shortest path between one node to another on Wordnet. Leacock & Chodorow (1998) searched on the WordNet synset network using a breadth-first search to find out the shortest path between two concepts. The authors proposed the following equation for computing the semantic distance between two concepts c_1 and c_2 in WordNet:

$$sim(c_1, c_2) = -log \frac{len(c_1, c_2)}{2 \times \max_{c \in WordNet} depth(c)} \tag{2.2}$$

where $len(c_1, c_2)$ is the length of the shortest path between two concepts c_1 and c_2 on the WordNet tree.

Similarly, Wu & Palmer (1994) introduced a scaled metric for what they called conceptual similarity between a pair of concepts c_1 and c_2 in a hierarchy as:

$$sim(c_1, c_2) = \frac{2 \times depth(par(c_1, c_2)}{len(c_1, par(c_1, c_2) + len(c_2, par(c_1, c_2)) + 2 \times depth(par(c_1, c_2))} \quad (2.3)$$

where $par(c_1, c_2)$ is the nearest parent node of c_1 andc_2 on the tree.

These works based on the nature structure of WordNet and its well defined knowledge have offered reliable measures on semantic distance. However, these methods also suffer from coverage problems with WordNet. The uses of WordNet are relatively sufficient for particular limitted domains. When it comes to apply in diverse text domains such Web texts or medical texts, the WordNet vocabulary is relatively small and limited to cover most of new terms and concepts.

There have been several proposals to address the issue of WordNet by using corpus statistics about the nodes and weighting the links according to some measurement over the node frequency statistics (Resnik 1995, Lin 1998*b*, Jiang & Conrath 1997).

Resnik (1995) proposed an alternative way to measure semantic similarity between words using corpus statistics and lexical taxonomy. That is the more information two concepts shared in common, the more similar they are, and the information content of the concepts that subsume them in the taxonomy. The similarity between two

concepts was defined as follows:

$$sim(c_1, c_2) = \max_{c \in S(c_1, c_2)} \{-\log p(c)\} \tag{2.4}$$

where $S(c_1, c_2)$ was the set of concepts that subsume both c_1 and c_2, and $p(c)$ as function $p : C \mapsto [0, 1]$, such that for any $c \in C$, $p(c)$ is the probability of encountering an instance of concept c. This implies that p is monotonic as one moves up the taxonomy: if c_1 IS-A c_2, then $p(c1) \leq p(c2)$. Moreover, if the taxonomy has a unique top node then its probability is 1. In the case of multiple inheritances, where words have more than one sense, word similarity is determined by the best similarity value among all respective pairs of senses as follows:

$$sim(w_1, w_2) = \max_{c_1 \in sense(w_1)} \max_{c_2 \in sense(w_2)} \{sim(c_1, c_2)\} \tag{2.5}$$

where $sense(w_1)$ denotes the set of possible senses for word w.

Lin (1998*b*) attempted to define a measurement of similarity between words using information commonality. The similarity between A and B was measured by the ratio between the amount of information needed to state their commonality and the

information needed to fully describe what they are:

$$\begin{aligned} sim(c_1, c_2) &= \frac{\log p(comm(c_1, c_2)}{log(p(descr(c_1, c_2)))} \\ &= \frac{2 \times log(p(parent(c_1, c_2)))}{\log p(c_1) + \log p(c_2)} \end{aligned} \tag{2.6}$$

where $parent(.)$ and $p(.)$ were defined as Equations 2.3 and 2.4 respectively.

Recently, Agirre et al. (2009) have studied a semantic similarity approach using WordNet. Given a pair of words and a graph-based representation of WordNet, the authors first computed the personal PageRank over WordNet separately for each of the words, producing a probability distribution over WordNet synsets. Secondly, each word was represented by a vector of its discrete probability distributions. The similarity between two words was the Cosine distance between the two respective vectors.

Wikipedia (Wikipedia 2004), on the other hand, is an universal encyclopaedia of knowledge, which is manually created by contributors over the world. As written in different languages, Wikipedia is considered as the world of knowledge, which has also inspired researchers in most fields to discover its potentials. The richness and diversity of this repository have been explored and used in a substantial amount of work on semantic analysis (Yeh et al. 2009, Gabrilovich & Markovitch 2007, Strube & Ponzetto 2006, Hassan & Mihalcea 2011, Radinsky et al. 2011, Zesch et al. 2008)

Strube & Ponzetto (2006) proposed a method to measure the semantic relatedness

between words namely WikiRelate!. The method aimed to overcome the limitation of WordNet by using a larger knowledge repository, Wikipedia, to increase its coverage. The relatedness between a pair of words was measured by using information from Wikipedia pages as well as the shortest path was calculated using the category structure of Wikipedia.

In the work of Gabrilovich & Markovitch (2007) (see 2.2.5), ESA was proposed to measure the semantic relatedness. The intuition behind word representation from ESA was quite simple. Each Wikipedia document was considered as an explicit concept (topic), and the representation of a word was a distribution over entire Wikipedia concepts. Given a word w_i extracted from Wikipedia articles d_j $(j = 1 \dots n)$, the representation of w_i was measured by a vector of weighted concepts. The weighting matrix reflected the importance of the relation between word w_i and the concept d_j as follows:

$$T(w_i, d_j) = tf(t_i, d_j) \times \log \frac{n}{df_i} \tag{2.7}$$

where $T(w_i, d_j)$ was the weighting which reflects the important associations between the word w_i, df_i was the document frequency of w_i, and $tf(t_i, d_j)$ was defined as

$$tf(t_i, d_j) = \begin{cases} 1 + \log count(t_i, d_j), & if \ \ \log count(t_i, d_j) > 0 \\ 0 & otherwise \end{cases} \tag{2.8}$$

In the situation that Wikipedia links were taken into account, the new weighting

$T^*(w_i, d_j)$ was measured as:

$$T^*(w_i, d_j) = T(w_i, d_j) + \sum_{\{j|\exists link(d_j, d_i)\}} T(w_i, d_j) \tag{2.9}$$

where $link(d_j, d_i)$ is represented for a link from document d_j to document d_i using a Wikipedia link.

Once the representation of each word was constructed as a vector of Wikipedia concepts, the similarity between a pair of words was measured by the Cosine distance between their respective vectors. Inherited the work from ESA, Temporal Semantic Analysis (TSA) was proposed by extending ESA with time series information to adjust the feature weighting as well as adding more external concepts from social media (Radinsky et al. 2011).

Similar to ESA, Hassan & Mihalcea (2011) proposed Salient Semantic Analysis (SSA). The method used the model of representing word features by Wikipedia concepts, but instead of using existing Wikipedia concepts and links to model features and weightings, the authors focused on only existing Wikipedia concepts while extracting salient concepts under the Wikipedia text. Given a list of entire Wikipedia concepts, the SSA method labeled salient concepts by matching the Wikipedia text to the list of Wikipedia concepts. Once the list of Wikipedia concepts and silent concepts were identified, the co-locations between the concepts and words surrounding in a window size were identified to construct the words' semantic representation. In

other words, given a word extracted from a Wikipedia text, its semantic representation was viewed as a vector of concepts where each feature weighting was identified by the co-location between them within a window size.

Formally, given a corpus C with vocabulary size N, and concept size W (number of unique Wikipedia concepts), a co-occurrence $N \times W$ matrix (E) was generated representing the accumulative co-occurrence frequencies of each of the corpus terms with respect to its contextual concepts (defined by a context window of size k). The elements of E were defined as follows:

$$E_{ij} = f^k(w_i, c_j) \tag{2.10}$$

where f^k was the number of times the terms w_i and concept c_j co-occur together within a window of k words in the entire corpus. The matrix was further processed to generate an $N \times W$ PMI matrix P, with elements defined as:

$$P_{ij} = P(w_i, c_j) = \log_2 \frac{f_k(w_i, c_j) \times m}{f^C(w_i) \times f^C(c_j)} \tag{2.11}$$

where $f^C(w_i)$ and $f^C(c_j)$ were the corpus frequencies for the term w_i and concept c_j respectively.

Each row P_i was further filtered to eliminate irrelevant associations by only keeping the top β_i cells and zeroing the rest. This corresponds to selecting the β_i highest

scoring PMI terms associated with a given row:

$$\beta_i = (log_{10}(f^C(w_i)))^2 \times \frac{log_2(N)}{\delta}, \delta \geq 1 \tag{2.12}$$

where δ was a constant that is adjusted based on the size of the chosen corpus. To calculate the semantic relatedness between two words given the constructed matrix, the Cosine distance measurement was adopted on two respective vectors.

In other related work, Yeh et al. (2009) proposed WikiWalk, a method to measure the semantic relatedness using Random walks on the Wikipedia repository. WikiWalk was conducted by converting Wikipedia into a graph, mapping input texts into the graph and performing random walk based on Personalized PageRank to obtain stationary distributions that characterize each text. The semantic relatedness between two texts was computed by comparing their distributions.

2.4.2 Content-based Approaches

In contrast to KB approaches, content-based (CB) approaches aim to utilise the amount of unstructured text available. One of the advantages of these methods is that they can operate independently on different text domains and languages. While no knowledge-based structures are used, the CB approaches rely only on the analysis of text content such documents, paragraphs, sentences, or a chunk of text. Their focuses are to disclose the semantic signatures within the unstructured text content

and transfer these into a presentation that can disclose their meanings.

Some research works have investigated on the large amount of free text available, and come up with semantic representation of words such as Schutze & Pedersen (1993), Dumais (2004), Lin (1998*b*), Collobert et al. (2011), Dagan et al. (1993), Dinu & Lapata (2010), Huang et al. (2012), Reisinger & Mooney (2010), Sahlgren (2006) and Turney (2001)

Dagan et al. (1993), for example, proposed a method to measure the similarity between words using unobserved data. This method was based on the assumption that similar word co-occurrences have similar values of mutual information, and therefore the similarity measurement was designed using the similarity metric between two vectors of mutual information values. Given a pair of words (w_i, w_j), which co-occurs no more than a distance d words in a text corpus, mutual information about the co-occurring pair $I(w_i, w_j)$ was estimated by:

$$I(w_i, w_j) = \log_2(\frac{N}{d}\frac{f(w_i, w_j)}{f(w_i)f(w_j)}) \tag{2.13}$$

where N was the number of words in the corpus and $f(w_i, w_j)$ was the count of (w_i, w_j) in the corpus.

To measure the semantic similarity between two words (w_i, w_j), the authors utilised indirect co-occurrence with another w on the corpus. The co-occurrences between words were directional with the word collocations on the left and the right were also

considered. Thus, the semantic similarity between words was measured as:

$$sim(w_i, w_j) = \frac{\sum_w sim_L(w_i, w_j, w).W_L(w_i, w_j, w) + \sum_w sim_R(w_i, w_j, w).W_R(w_i, w_j, w)}{\sum_w W_L(w_i, w_j, w) + W_R(w_i, w_j, w)} \tag{2.14}$$

where $sim_L(w_i, w_j, w)$ was defined as the left context similarity of w_i and w_j relative to w, and the $sim_R(w_i, w_j, w)$ was the right context similarity of w_i and w_j relative to w. These measurements were calculated as follows:

$$sim_L(w_i, w_j, w) = \frac{min(I(w, w_i), I(w, w_j)}{max(I(w, w_i), I(w, w_j))} \tag{2.15}$$

$$sim_R(w_i, w_j, w) = \frac{min(I(w_i, w), I(w_j, w)}{max(I(w_i, w), I(w_j, w))} \tag{2.16}$$

Schutze & Pedersen (1993) introduced context digests to form a high-dimension real-value representation of word meanings using typical left and right contextual words. The context digests summarize both how typical words appear as neighbours and how well they are substitutable for each other. The word meaning representation was created by encoded the word collocation matrix into a low-dimensional space of approximate matrix using singular matrix decomposition. The Cosine distance between two rows of the matrix was used to indicate how related of the semantic references between two words.

Lin (1998*b*) presented a semantic similarity measurement between words based on the commonality of patterns obtained between respective words. The author utilised

a syntactic parser to analyse textual information. Given a lexical syntactic pattern (w_i, r, w_j), where w_i, w_j are words in the text corpus and r is a lexical syntactic pattern represented for the relation between w_i and w_j, the information value of (w_i, r, w_j) was defined as:

$$I(w_i, r, w_j) = \log \frac{\parallel w_i, r, w_j \parallel \times \parallel *, r, * \parallel}{\parallel w_i, r, * \parallel \times \parallel *, r, w_j \parallel} \tag{2.17}$$

where $\parallel w_i, r, w_j \parallel$ was frequency of the pattern in the entire text corpus. Let $T(w_i)$ be the set of pairs (r, w_j) such that $I(w_i, r, w_j)$ was positive. The similarity between (w_i, w_j) was estimated as follows:

$$sim(w_i, w_j) = \frac{\sum_{(r,w) \in T(w_i) \cap T(w_j)} (I(w_i, r, w) + I(w_j, r, w))}{\sum_{(r,w) \in T(w_i)} I(w_1, r, w) + \sum_{(r,w) \in T(w_j)} I(w_j, r, w)} \tag{2.18}$$

Turney (2001) proposed a method to measure the semantic similarity between words using text available on a corpus. The intuition behind this work was that meanings of a word could be disclosed by its accompanying text. By investigating the relationship between a word and its surrounding neighbours, the distance between the meanings of two words was measured a point-wise mutual information value.

$$sim(w_i, w_j) = \log_2 \frac{p(w_i, w_j)}{p(w_i)p(w_j)} \tag{2.19}$$

where $p(w_i, w_j)$ was the probability that w_i and w_j co-occur.

Dumais (2004) measured the semantic similarity between words using their hid-

den topic representations. Given a matrix of word-document co-occurrences from a particular text corpus, the authors induced the semantic representation of words by applying Singular Vector Decomposition on the matrix. Formally, a $t \times d$ matrix of words and documents, X, was decomposed approximately into the product of three other matrices:

$$X = T_0 S_0 D_0^T \tag{2.20}$$

such that T_0 and D_0 have orthonormal columns and S_0 is diagonal. If the singular values in matrix S are ordered by size, the first k largest values may be kept and the remaining smaller ones set to zero. The product of the resulting matrices is a matrix $\hat{X}$ which was approximately equal to X as follows:

$$X \approx T_k S_k D_k^T \tag{2.21}$$

It can be shown that the new matrix $\hat{X}$ is the matrix of rank k and is closest in the least square sense to X. A co-occurrence representation of a word from the matrix X was now converted into the respective vector of hidden topics in X. In this case, the similarity between two words was estimated by the distance between two respective vectors in X.

Dinu & Lapata (2010) presented a method for measuring the similarity between words using latent factors. Each word in a given corpus is represented by a vector of latent senses. Assuming that a target word w_i found in a corpus share a global set of

latent senses $\{s_k|k:1\ldots K\}$, the representation of a word was modelled as follows:

$$v(w_i) = \langle\ p(s_1|w_i), ..., p(s_K|w_i)\ \rangle \tag{2.22}$$

where $p(s_j|w_i)$ was the probability of sense s_j given target word w_i. The similarity between two words was estimated by using the standard Cosine distance measurement on the respective vectors.

Vector space models have been popularly used for word representation and semantic distance measurement. Bengio et al. (2006), Mnih & Hinton (2007) and Collobert et al. (2011) automatically induced word representation to measure semantic similarity between words. Starting from a random initiated vector representation of each word (a vector of parameters), the representation of each word was learned and updated over input text examples extracted from a large amount of unstructured text data. All of these authors used deep learning architecture to induce word representation. Huang et al. (2012) enhanced the feature learning model to measure the semantic similarity between words. The authors took into account words in a window-size context as the local factor as well as a document context as the global factor to enrich the learning model. The semantic similarity between words was estimated by using the standard Cosine distance between two respective learned vectors.

2.5 Conclusion

In this chapter, we have presented the background theories to meaning representation in contexts as well as that for semantic distance measurement. We have also discussed various approaches that generate feature representation from different aspects of information. We has provided an overview of related works on measuring semantic distance under two categories: knowledge-based approaches and content-based approaches. The analysis presented in this chapter would provide the basic understandings used in the chapters that follow.

Chapter 3

RELATIONAL FEATURE ANALYSIS FOR SEMANTIC INTERPRETATION

This chapter presents a method for relation-based feature analysis for semantic interpretation. The method generates feature representation of a word by considering its relations extracted from local contexts as well as the hidden factors built from the sets of relations. The generated features of word meaning is evaluated based on the task of measuring semantic distance. Experimental results demonstrate the promising capacity of the relation-based features in modelling word meaning compared to traditional context-based features when tested on the same benchmarks.

3.1 Introduction

Understanding meanings of words is one of the key challenges in many language-based applications from document understanding, text summary to sentiment analysis. One of the reasons which makes this task harder is that the meanings not only depend on the way the words appear in local contexts, but also their usages may change over various genres and domains.

First, in the distributional hypothesis of Harris (1954), meanings of a word can be inferred from surrounding contexts where the word appears. Consider the following

example describing the contexts of an unknown word, "*tezgüino*", (Lin 1998*b*):

A bottle of *tezgüino* is on the table.

Mexicans like *tezgüino*.

Strong *tezgüino* makes you drunk.

We make *tezgüino* out of corn.

The contexts in which the word "*tezgüino*" appears suggest that "*tezgüino*" may mean a kind of alcoholic drinks made from "*corn*", that it gets people "*drunk*" and is normally contained in 'a '*bottle*". In other words, meanings of the word are disclosed by considering its relationships with other surrounding words in local contexts.

From a linguistics perspective, word meanings can be identified from local contexts where the *syntagmatic relations* and *paradigmatic relations* play important roles. They jointly describe the different aspects of a word's meanings (Sahlgren 2006). While paradigmatic relations support the meanings in term of long distant relations, the syntagmatic relations govern the meanings of the word in term of how it interacts with its adjacent neighbours. Words share in a paradigmatic relation as long as they are inter-changeable in their contexts but still maintain similar meanings in the contexts. For instance, the word *tezgüino* in the contexts above could be exchanged for any word on holding the meanings of "*alcoholic drink*" as they share the same lexical patterns such as "*strong * makes you drunk*[1]" in the local contexts. Syntagmatic re-

[1]* is represented for any word that does not change meanings of the sentence

lations are different as they describe the property/attribute features. Words such as "*bottle*" and "*corn*" are considered attributes of "*tezgüino*". The roles of syntagmatic and paradigmatic relations again emphasize the importance of relationships between a word and its surroundings in local contexts.

Additionally, genres and domains where a word appeared also contribute to determining meanings of a word. Consider the word "*banks*" in the following example sentence:

The grade of river *banks* can vary from a vertical to a shallow slope.

The conceptual meanings of the word "banks" can be found by analysing the conceptual contexts where it appears. In this case, its meanings refer to the entity "*water body*". Moreover, the meanings of a word are also guided by the domains in which it belongs. In the above example, the meanings refer to the concept of "*geography*" rather than that of "*financial institution*".

Previous work in the field considers the issue of a word in its local contexts and its conceptual meanings in different ways. On the one hand, a word in contexts is analysed by considering its relationships with surrounding neighbours using a sliding window context (Agirre et al. 2009), or a lexical pattern (Lin 1998*b*, Turney 2006). On the other hand, word meanings are discovered by analysing its associated concepts (Dumais 2004, Dinu & Lapata 2010, Gabrilovich & Markovitch 2007). In this work, we propose a method of feature analysis for semantic interpretation, which

takes into account the relationship of words in local contexts as well as conceptual information associated with the focus word. The method first analyses a word in its local contexts to extract the relevant semantic relations. Secondly, any hidden concepts are disclosed by a process of inducing the extracted relations. Finally, the meaning representation of the word is constructed as a high dimensional data point in a hidden feature space, which is then evaluated in term of the semantic distance measurement.

The rest of this chapter is as follows: in Section 3.2, we first introduce a method to construct the semantic space for a word using relations in context. Section 3.3 then introduces how the semantic space of a word is constructed. In Section 3.4, we present latent features for semantic representation, which are constructed by using a matrix factorization technique. Experiments based on various approaches to feature generations are presented in Section 3.5. Section 3.6 discuss the evaluations on standard testing benchmarks. Finally, we conclude the chapter in Section 3.7.

3.2 Meaning Representation Using Relations in Context

As discussed in previous examples, the relationships between a word and its surroundings play an important role in describing word meanings. Given a word in context, neighbouring words will be identified if a relation exists between the neibouring words and the target word. Consider the previous example where the meanings of the word "*tezgüino*" can be inferred by its relationships with surrounding words as follows:

(bottle, of, *tezgüino*)

(*tezgüino*, is on, table)

(Mexicans, like, *tezgüino*)

(Strong, , *tezgüino*)

(*tezgüino*, makes you, drunk)

(*tezgüino*, made out of, corn)

The meanings of the word "*tezgüino*" can be understood via multiple relations that it appears. A relation is described as a triple which reflects the relationship between a word and its neighbour using a lexical relation in a middle. Formally, a relation between a word and its neighbour is defined as a triple (e, r, e') where e and e' are denoted as an entity and r is defined as a predicate between e and e'. Below, we describe an extraction algorithm, which extract relations of words in local contexts.

3.2.1 Relation Extraction Algorithm

The relation extraction algorithm takes part-of-speech tags of a sequence of words as the input to extract a set of triples (e, r, e'). The algorithm is designed to single-pass through the sequence and extract triples using the following phases:

1. Entity Identification: As a triple contains the relationship between an entity and a neighbour one, one of the entities has the subject role in the relationship, which is as a single noun or a name entity. The other entity expresses an object

role in the relationship which is as a single noun, name entity, or adjective displaying the characteristics of the first entity.

2. Predicate Identification: The predicate is as the relational phrase connecting two entities. It also asserts a semantic relationship between them. The sequence of words appearing in-between two entities is extracted as long as it satisfies the following lexical syntactic constraints:

$$\mathbf{V+ \mid V+W*P \mid P}$$

where:

V= (relative word | verb | particle | adverb)

W = (noun | adjective | adverb | pronoun | determiner)

P = (preposition | particle | appositional modifier)

3. Confidence Score Assignment: Given components of a triple identified from a sequence of text, a confidence model will be applied to assign a confidence value for the triple. The confidence value is then used to determine what are the unreliable extractions.

The relation extraction algorithm takes an unstructured input text and returns a set of extracted triples. With the complexity of unstructured text data, information

from triples can be in various forms. In the next section, we design a confident model which facilitates the selection of reliable and non-reliable relations between words.

3.2.2 Confidence Function for Extracted Triples

One of the advantages of the proposed extraction algorithm is that it quickly travels through the sequence of a text to identify the triples. It is able to produce large amounts of triples from a given text corpus. However, due to the complexity of English language structure as well as the mis-identification of the POS tagger, there is a need to leave out the mis-identified relations before constructing meaning representation using the extracted triples. For doing so, a binary classification model is implemented, which classifies a triple into reliable and non-reliable classes. The model takes an extracted triple as an input and returns a confidence value which is then compared to a given threshold to identify whether the triple is reliable.

Given a triple extracted from a text sequence, it is then transformed into a vector representation using the list binary features as in the Table 3.1. The classification model is then trained using logistic regression on the training example sentences, which are parts of the extractions from the OpenIE[2] and manually labelled as either reliable or non-reliable relations.

Given $x = (x_1, x_2, \ldots, x_n)$ as a vector representation of a triple t, $p(x)$ as the confidence function of x, the model of logistic regression that describes the confidence

[2]http://reverb.cs.washington.edu/reverb_clueweb_tuples-1.1.txt.gz

Table 3.1: List of features used to identify whether a triple is reliable or un-reliable. These features was inherited from the work of Fader et al. (2011) in Open Information Extraction.

Features	Description of Features
x_1	$t = (e_i, r_{ij}, e_j)$ covers all words in the sequence s
x_2	The last word in r_{ij} is *for*
x_3	The last word in r_{ij} is *in*
x_4	The last word in r_{ij} is *of*
x_5	r_{ij} satisfies the lexical syntactic constraints
x_6	The last word in r_{ij} is *to*
x_7	s begins with e_i
x_8	s ends with e_j
x_9	$\text{len}(r_{ij}) \leq 6$
x_{10}	The last word in r_{ij} is *on*
x_{11}	e_i is a proper noun
x_{12}	e_j is a proper noun
x_{13}	r_{ij} starts with a relative WH-word
x_{14}	Coordinate conjunction to the right of e_j in s
x_{15}	There is a preposition to the left of e_i in s
x_{16}	$len(t) \geq 10$ words
x_{17}	There is a NP to the right of e_j in s
x_{18}	The last word in t_{ij} is a conjunction

function is as follows:

$$\log \frac{p(x)}{1 - p(x)} = \beta_0 + \sum_{k=1}^{n} \beta_k . x_k \tag{3.1}$$

where $\beta = (\beta_0, \beta_1, \ldots, \beta_n)$ is a vector of the parameters. We extend x with $x_0 = 1$,

Equation 3.1 turns into:

$$\log \frac{p(x)}{1 - p(x)} = \sum_{k=0}^{n} \beta_k . x_k \tag{3.2}$$

$$= \beta^T x \tag{3.3}$$

Solving (3.2) for p, this gives the confidence function of x given a parameter vector β as

$$p(x;\beta) = \frac{1}{1+e^{-\beta^T x}} \tag{3.4}$$

Given a set of training examples $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$, the vector of parameter β is estimated using the logistic regression algorithm. Once the parameters in β are estimated, the confidence value of a triple is identified using the Equation 3.4.

3.2.3 Learning Parameters for the Confidence Function

For a training example (x, y), where x is a vector representation of a triple and y is the actual class of a triple (1 = reliable class; 0 = non-reliable class), the logistic regression cost function as a penalty to predict the class for the training example (x, y) is defined as:

$$cost(p(x;\beta), y) = \begin{cases} -\log p(x) & if \quad y = 1 \\ -\log\left(1 - p(x)\right) & if \quad y = 0 \end{cases} \tag{3.5}$$

This can be rewritten as:

$$cost(p(x;\beta), y) = -y \log p(x) - (1-y)\log\left(1 - p(x)\right) \tag{3.6}$$

Given all training examples $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$, the cost function for these ex-

amples is defined as:

$$L(\beta) = \sum_{i=1}^{m} cost(p(x^{(i)}; \beta), y^{(i)}) \tag{3.7}$$

$$= -\sum_{i=1}^{m} y^{(i)} \log p(x^{(i)}) + (1 - y^{(i)}) \log (1 - p(x^{(i)})) \tag{3.8}$$

Solving this for β gives

$$L(\beta) = -\sum_{i=1}^{m} -log(1 + e^{\beta^T x}) + \sum_{i=1}^{m} y^{(i)} \beta^T x \tag{3.9}$$

To fit the parameters β, we need to find parameters that minimizes the cost function. In order to do this, we use gradient descent to update the parameters for every single training example $x^{(i)} = (x_0^{(i)}, x_1^{(i)}, \ldots, x_n^{(i)})$ as follows:

$$\beta_k := \beta_k - \alpha \frac{\delta}{\delta \beta_k} L(\beta) \tag{3.10}$$

where α is the learning rate, and $\frac{\delta}{\delta \beta_k} L(\beta)$ is the derivative of the cost function with respect to one component of β such as (β_k). The partial derivative respects to β_k is calculated as:

$$\frac{\delta}{\delta \beta_k} L(\beta) = \sum_{k=0}^{n} (p(x^{(i)}; \beta) - y^{(i)}) x_k^{(i)} \tag{3.11}$$

and therefore, each component of the parameter β can be updated simultaneously for

each training example as in the following:

$$\beta_k := \beta_k - \alpha \sum_{k=0}^{n} (p(x^{(i)}; \beta) - y^{(i)}) x_k^{(i)} \tag{3.12}$$

Once the parameters in β have been estimated using all of training examples, the confidence function 3.4 is used to estimate the confidence value of each relation.

3.2.4 The relational Semantic Space of Words

Using local context information, the meanings of a word are identified by considering its relationships with surrounding words. Once triples associated with a word are extracted using the extraction algorithm, the semantic space of the word is constructed by encoding extracted relations into a vector space model.

Given w_i as a focus word and its associated triples as $\{t^{(1)}, t^{(2)}, \ldots, t^{(N)}\}$, each triple $t^{(j)}$ is encoded into the semantic space of w_i as a vector feature w_j with the constraints that both w_i and w_j are two entities of the triple $t^{(j)}$. In this case, weighting of the feature w_j is estimated by the association degree between w_i and w_j within the triple $t^{(j)}$. Formally, the semantic space of w_i is constructed as a sparse vector space model as follows:

$$v(w_i) = (w_1 : wt(t^{(1)}), w_2 : wt(t^{(2)}), \ldots, w_N : wt^{(N)}) \tag{3.13}$$

where $wt(t^{(j)})$ is association degree between the entity w_i and w_j within the triple

$t^{(j)}$, which is estimated as:

$$wt(t^{(j)}) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)} \tag{3.14}$$

$$p(w_i, w_j) = \frac{d(w_i, w_j)}{\sum_{i,k=1...n} d(w_i, w_k)} \tag{3.15}$$

$$p(w_i) = \frac{\sum_{k=1...n} d(w_i, w_k)}{\sum_{i,k=1...n} d(w_i, w_k)} \tag{3.16}$$

where $d(w_i, w_k)$ is the number of times that w_i and w_k co-occur.

As a summary, this section has presented a method for modelling word meanings using relations extracted from local contexts. The method utilises the relation extraction technique to combine with a confident value filter to select reliable features for meaning representation. With the expectations to see the effects of different feature filters on meaning representation, we conduct a NGram-based features representation of words using various feature filters. Details of this work are presented in the next section.

3.3 Meaning Representation Using N-Gram Context

Different from the argument that meanings of a word are best expressed only using highly associated relations within local contexts, word representation can be accumulated by considering its possible collocation associations with neighbouring words within local contexts. The idea is quite straightforward and it has made a significant

contribution to express the word meanings (Harris 1954) and to measure semantic similarity (Agirre et al. 2009). In this section, we present a semantic representation of words using the idea of distribution hypothesis. However, we have updated the weighting models and weighting filters, which highlight the effects of information values on word meaning representation.

3.3.1 Building the Word Semantic Space Using NGram Context

The semantic space of a word was constructed by using as much as possible the local context information where the word appeared. Given a focus word in a text corpus, we applied a NGram extraction technique to extract all possible collocation words appearing in a given sliding window for the focus word. The extracting results over the entire text corpus were then encoded into a vector space model to build a semantic space for the focus word. Formally, given w_i as the focus word and $w_1, w_2 \dots w_N$ as the extracted collocation words within a given NGram context, the semantic space of the focus word is represented as follows:

$$v(w_i) = (w_1 : wt_{i1}, w_2 : wt_{i2}, \dots, w_N : wt_{iN}) \tag{3.17}$$

where wt_{ij} is the weighted collocation association between w_i and w_j within the given NGram context. wt_{ij} was also measured using PMI as follows:

$$wt_{ij} = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)} \tag{3.18}$$

3.3.2 Weighting Filters

Section 3.3.1 presented the representation of words using surrounding words within a sliding window context. However, due to the sparseness of the vector representation, we needed to design two weighting filters to reduce this sparseness as well as the low information value of the representation as discussed below:

The first filter, namely the frequency filter (FF), aimed to remove non-essential associations between a pair of collocation words. The FF filter only retained the features if the frequency of the collocation pairs in the corpus greater than FF. The value of FF was essential in eliminating rare words as features in the vector representation.

In using the second filter, we expected to see the influence of low information value extractions on the effectiveness of meaning representation. We applied a filter to the weighting value of each extraction, which is called an information value filter (IVF). Only a feature that had its weighting equal or greater than IVF would remain in the vector representation, otherwise it was set to 0. The effects from using the IVF immediately eliminate the contribution of highly frequent features as well as any rare features. Once all the filters had been applied, the final vector representation of

NGram-based features was constructed.

3.4 Inducing Hidden Concepts from Semantic Space

Word meanings in context can be represented by a semantic space, which takes into account the semantic relations between a word and its surroundings. Moreover, word meanings are also influenced by the global genres or concepts it implicitly belongs to. In this section, we present the application of a hidden concept analysis on semantic space words. The analysis takes into account word-word interaction within hidden concepts to figure out the representation of words.

In a sparse matrix where each row of the matrix represent the semantic space of a word, the matrix is analysed by single value decomposition to derive particular hidden semantic structures, called latent concepts. Let $A = w \times f$ be a sparse matrix of words w and features f. The matrix A can be decomposed into the product of three matrices.

$$A = W \circ S \circ F^T \tag{3.19}$$

where W, S, and F^T are full-rank matrices. While W and F are orthogonal matrices, S is diagonal matrix of singular values.

When singular values in S are ordered by size, the first k largest singular values are kept and the remaining smaller values set to zero; the product of the three matrices is an approximation of A and is called $\hat{A}$. Since zeros are introduced in S, the representation can be simplified by deleting the zero rows and columns of S to obtain

S_k with rank k, and then deleting the corresponding columns of W to obtain W_k and rows of F to obtain F_k. This results in a reduced model as follows:

$$A \approx \hat{A} = W_k \circ S_k \circ F_k^T \tag{3.20}$$

In terms of semantic interpretation, the dot product between two row vectors of $\hat{A}$ reflect the extent to which two words have similar patterns of weighting across the set of features. Since S is diagonal and T is orthonormal, it can be verified that:

$$\hat{A}\hat{A}^T = W_k \circ S_k \circ S_k \circ W_k^T \tag{3.21}$$

The matrix AA^T is a square symmetric matrix containing all these word-to-word dot products. This means that the cell $[i, j]$ of $\hat{A}\hat{A}^T$ can be obtained by taking the dot product between the i and j rows of the matrix $W_k S_k$. Each row of the matrix $W_k S_k$ can also be considered as the representation of a respective word. In order words, given a vector representation of a word in matrix A, its vector representation using hidden concepts as features is the corresponding row in the matrix $W_k S_k$.

In summary, given a matrix of words and their relational features, the output of matrix factorization is the matrix of words and their *relation-based hidden features*, which can also be considered as the representation of words using both relational features in local context as well as its global domains. Similarly, using the matrix of words and their NGram word features as the input, the matrix factorization returns

a matrix of words and *NGram-based hidden features* for the semantic representation.

3.5 Experiment

3.5.1 Testing Benchmarks

We used two popular semantic distance benchmarks WS-353 (Finkelstein et al. 2001) and RG-65 (Rubenstein & Goodenough 1965) to evaluate the effectiveness of the proposed features for semantic representation. We also used the MTurk (Radinsky et al. 2011) dataset as an independent test for scanning parameters (see Section 2.3.1 for more details about these datasets).

3.5.2 Corpus

Traditionally, content-based methods are operated on unstructured text repositories. In our experiments, we simply used a text corpus containing large amounts of text information, which also included a variety of texts from different genres. Thus, Wikipedia text was selected for the experiments as it is one of the largest text repositories freely available on the Web. In particular, we used Wikipedia English XML dump of October 01, 2012. After parsing the XML dump using Wikiprep[3], The first 1,000,000 articles were used for the experiments.

[3]Wikiprep http://sourceforge.net/projects/wikiprep/

3.5.3 NGram Word Features Extraction

Given a piece of text from the Wikipedia article, we used the NGram algorithm (sliding window $ws = 3$) to extract pairs of collocation words within a sliding window. The extraction results from the entire text corpus were normalised by a stemming technique (Van Rijsbergen et al. 1980) before applying the FF filter. The retained extractions were weighted using the PMI weighting algorithm before applying the IVF filter to return the final semantic space for a given word.

As the influences of FF and IVF are significant in the representation of words as well as the task of semantic distance measurement. We conducted an independent semantic distance test on the MTurk dataset to select the values of FF and IVF. The best Spearman's correlation score $\rho \approx 63\%$ was obtained on the NGram word features with $FF = 2$ and $IVF = 1$. This correlation value was comparable to other methods that were tested using the same MTurk dataset. Table 3.2 shows detailed information on how FF and IVF were selected from the independent test.

Table 3.2: Experiment using MTurk for turning parameters. The best Spearman's correlation score (%) was obtained with $FF = 2$, $IVF = 1$ on NGram word features, and with $r = 0.7$ on the confident threshold. Results of the related works on the same dataset were also presented.

Algorithm	$\rho \times 100$
Explicit Semantic Analysis (Radinsky et al. 2011)	59
Temporal Semantic Analysis (Radinsky et al. 2011)	63
NGram Word features	**62.84**
Relational features	61.47
Relational features with confidence ($r = 0.7$)	64.54

Once the semantic space of a word was constructed using all of its associated triples, the semantic representation of the word was also created by encoding the semantic space into a sparse vector of NGram word features. The effectiveness of this representation was also evaluated on the task of the semantic distance measurement.

3.5.4 Relational Feature Extraction

Triples were extracted using a pattern-based method as presented in Section 3.2.1. Given a piece of text from a Wikipedia article, the designed heuristic rules single pass through the text and return the extractions. From all of the selected Wikipedia articles, 53,653,882 raw unique triples were obtained . Each component of the triples was then normalized by a stemming technique (Van Rijsbergen et al. 1980). The frequency filter for each triple ($FF = 2$) was also applied with 47,143,381 unique triples being obtained after normalization. The extractions associated with each word were then accumulated and weighted using PMI weighting schema. To generate the semantic space of the word, the IVF filter was applied to remove low information value triples. The semantic representation of the word was also constructed by encoding its respective semantic space into a sparse vector of relational word features (see Section 3.2.4).

Table 3.3: List of features using their weightings updated from the Logistic Regression algorithm on training examples

Features	Weighting	Description of Features
x_1	0.88	$t = (e_i, r_{ij}, e_j)$ covers all words in the sequence s
x_2	0.77	The last word in r_{ij} is *for*
x_3	0.72	The last word in r_{ij} is *in*
x_4	0.69	The last word in r_{ij} is *of*
x_5	0.67	r_{ij} satisfies the lexical syntactic constraints
x_6	0.61	The last word in r_{ij} is *to*
x_7	0.59	s begins with e_i
x_8	0.55	s ends with e_j
x_9	0.54	$\text{len}(r_{ij}) \leq 6$
x_{10}	0.49	The last word in r_{ij} is *on*
x_{11}	0.41	e_i is a proper noun
x_{12}	0.33	e_j is a proper noun
x_{13}	0.25	r_{ij} starts with a relative WH-word
x_{14}	0.15	Coordinate conjunction to the right of e_j in s
x_{15}	-0.32	There is a preposition to the left of e_i in s
x_{16}	-0.35	$len(t) \geq 10$ words
x_{17}	-0.45	There is a NP to the right of e_j in s
x_{18}	-0.61	The last word in t_{ij} is a conjunction

3.5.5 Relational Feature Extraction with Confidence

The purpose of using a confidence model was a different mean of eliminating non-essential triples. It has contributed to the selection of highly reliable relations for building the semantic space. In order to train the model, we manually created training examples including 1000 reliable relations and 1000 unreliable relations respectively from the raw triples in the previous step. These examples would be used to train parameters for the confidence model.

To all of the triples after stemming from the previous step, the confidence function

(see Equation 3.4) was then applied to each triple to return its confidence value. The confidence threshold ($r = 0.7$) was then selected to retain those extractions with their confidence values greater than the threshold. After the training phrase, the weighting parameters of β have been updated and shown in Table 3.3. We obtained a total of 36,764,371 triples whose confidence values were greater than the given threshold. This meant more than 22% of the triples could be considered as unreliable using the given threshold. The retained extractions were accumulated and weighted to return the semantic space for each given word as a sparse vector of relational word features with confidence.

3.5.6 Hidden Feature Extraction

Given a list of entire words in the Wikipedia corpus, each word was then represented as a high dimensional vector of features (either relational features or NGram word features). The resulting set of vectors was encoded into a matrix where each row represented for the vector, and each column represented for the feature. The matrix was then approximated using matrix factorization with $K = 100$ as the number of singular values(see Section 3.4). In this way, each word was represented as a vector of hidden features. In the case where a set of vector of relational features with confidence was used as input, a set of vector of relation-based hidden features would be returned. Similarly, vectors of NGram-based hidden features would be created when vectors of NGram word features were used as the input data for matrix factorization. When

K singular values are used in the matrix factorization to generate K hidden features, each of the features is considered as the representation of a hidden topic. The value K = 100 was selected with a assumption that each given text will have a maximum of 100 hidden topics. This selection can only indicate the contribution of hidden topics in meaning representation, which can be changed depending on domains of applications. Discuss about the selecting a optimal K is beyond the scope of this work.

3.5.7 Semantic Distance

After the preprocessing step, five different kinds of vector representation were resulted:

- Relational features
- Relational features with confidence
- NGram word features
- Relation-based hidden features
- NGram-based hidden features

To evaluate the effects of the meaning representation of words, we implemented the task of the semantic distance measurement. Given $v(w_i)$ and $v(w_j)$ as the vector representation of two words w_i and w_j, their semantic distance was measured directly using the standard *Cosine* distance measurement as follows:

$$dist(w_i, w_j) = \frac{v(w_i) \times v(w_j)}{\|v(w_i)\| \times \|v(w_j)\|} \tag{3.22}$$

3.6 Evaluation

3.6.1 Overall results compared to other content-based methods

In this section, we discuss about the effectiveness of word representations using different feature generation methods over semantic distance datasets. Tables 3.4 and 3.5 show the experimental results of various features tested on WS-353 and RG-65 datasets, and the results from related methods tested on the same benchmarks.

Table 3.4: The experimental results (%) with different content-based methods on WS-353 datasets using Spearman's rank correlation (ρ). In the first row, results from the related work are presented. The results of our proposed features are showed in two last rows of the table.

Algorithm	$\rho \times 100$
Syntactic Features (Lin 1998*b*)	34.80
Latent Topic Features (LSA) (Finkelstein et al. 2001)	58.10
Latent Topic Features (LDA) (Dinu & Lapata 2010)	53.39
Single-Prototype (Reisinger & Mooney 2010)	55.3
Multi-Prototype (Reisinger & Mooney 2010)	**76.9**
Multi-Prototype (Huang et al. 2012)	71.3
Learned Features (Collobert et al. 2011)	49.86
Context Window Pattern (WS=1) (Agirre et al. 2009)	69
Context Window Pattern (WS=4) (Agirre et al. 2009)	66
NGram Word Features (WS=3, FF=2, IVF=1)	71.09
Relational Features (FF=2, IVF=1)	69.42
Relational Features with Confidence (FF=2, $r = 0.7$)	72.25
NGram-based Hidden Features (rank K=100, WS=3, FF=2, IVF=1)	70.61
Relation-based Hidden Features (rank K=100, FF=2, IVF=1)	68.25
Relation-based Hidden Features with Confidence (rank K=100, FF=2, $r = 0.7$)	71.16

The first section of the Tables 3.4 and 3.5 shows that all the proposed feature

Table 3.5: Experimental results (%) from various content-based methods on RG-65 datasets using Spearman's rank correlation (ρ). The first section shows the results from the related works. The results of our proposed features are also demonstrated in two last sections of the table

Algorithm	$\rho \times 100$
Syntactic Features (Lin 1998*b*)	78.8
Latent Topic Features (LSA) (Finkelstein et al. 2001)	60.9
Context Window Pattern (WS=1) (Agirre et al. 2009)	89
Context Window Pattern (WS=4) (Agirre et al. 2009)	**93**
NGram Word Features (WS=3, FF=2, IVF=1)	79.56
Relational Features (FF=2, IVF=1)	79.72
Relational Features with Confidence (FF=2, $r = 0.7$)	85.75
NGram-based Hidden Features (rank K=100, WS=3, FF=2, IVF=1)	79.38
Relation-based Hidden Features (rank K=100, FF=2, IVF=1)	77.63
Relation-based Hidden Features with Confidence (rank K=100, FF=2, $r = 0.7$)	85.31

generation methods outperform to the baseline method, LSA, and the work of Lin (1998*b*) on both datasets. The reason these works were selected for comparison was that the fundamental ideas embodied in these methods were quite similar to those presented in the current work. However, we used qualified relations extracted from a text rather than syntactic relations. Although inherited from LSA, the ideas of using matrix factorization to manipulate the semantic space of words, our work was more focused on extracting critical relations in context. This is in contrast to LSA where word collocated in contexts are used as an indicator to construct a matrix of word-word relations. Moreover, the ideas of using word-word collocation in contexts from LSA and Agirre et al. (2009) is also quite similar to the proposed NGram word features. However, we used PMI as the weighting mechanism rather than word co-

occurrences to conduct word representations. Additionally, although parts of our work were quite similar to the work from Lin (1998*b*), our method gained benefits from using particular reliable relations between text rather than any syntactic relations. The better performance of the proposed relational features was also inherited from the advance of confidence model, which eliminates noises from non-essential relations.

On the other hand, our proposed methods are relatively comparable to other content-based methods on both datasets, particularly, the WS-353 dataset (Table 3.4). Although the proposed methods outperformed most of the popular content-based methods on the task of measuring semantic distance, the results also show that the proposed methods are under comparable to those using multiple vector representation for word meanings, multi-prototypes (Reisinger & Mooney 2010). Similarly, the results from the RG-65 dataset also indicate that the proposed features improve the performance compared to other methods, although still lagging behind those achieved by Agirre et al. (2009).

3.6.2 Feature Generation Analysis

In terms of performance using the proposed features, the NGram word features returned a better correlation to human judgements than relational features in the case of applying weighting filters (FF, and IVF). In other words, this emphasizes the positive influences of IVF on the sliding window context compared to the relation-based context. However, the better performance of relational features using confidence model

Table 3.6: Correlation results (%) for NGram word features and relational word features when applying a frequency filter (FF=2) and information value filter (IVF) with the semantic distance tested on WS-353 dataset using Spearman's rank correlation (ρ). The best value is bolded and its corresponding parameter value is underlined

IVF	$\rho \times 100$	
	Relational word features	NGram word features
-3.0	60.58	58.95
-2.5	60.76	59.01
-2.0	61.05	59.32
-1.5	62.06	60.37
-1.0	63.49	62.39
-0.5	64.34	63.31
0.0	63.73	61.80
0.5	66.48	66.67
<u>1.0</u>	**69.42**	**71.09**
1.5	68.30	70.47
2.0	64.60	67.14
2.5	49.19	56.23
3.0	26.93	38.78

on both datasets indicates the effectiveness of eliminating features based on different lexical and syntactic criteria rather than those based on PMI weightings.

It is important to understand the influence of the filters on the overall performance. Table 3.6 shows in detail the affects of decreasing the amount of input data using IVF filter in relation to changing correlation values. Similarly, when selecting different confidence thresholds, the amount of data used for constructing the semantic space of words will also change which affects the performance on the semantic distance task in general. This is clearly showed in Table 3.7.

Using IVF or confidence threshold also brings different benefits and trade-offs.

While IVF focuses on the statistical information of a pair over the entire text corpus, a confidence model concentrates on the appearance of words syntactically in their local context only. Despite the straightforward feature generation methods proposed, one of the drawbacks is that the methods involved for representing words using high-dimensional vectors where the length of the vectors approximates the size of the entire word dictionary for the corpus. This requires a considerable amount of processing when integrating these representations into further applications.

Table 3.7: Correlation results (%) from relational features when applying different thresholds r on the confidence value for each triple with the semantic distance tested on the WS-353 dataset using Spearman's rank correlation (ρ). The best value bolded and its corresponding parameter value is underlined.

Confidence threshold r	$\rho \times 100$
0.1	60.58
0.2	61.15
0.3	63.05
0.4	65.16
0.5	68.28
0.6	69.14
<u>0.7</u>	**72.25**
0.8	69.37
0.9	65.12

To tackle the size of high-dimensional vectors, hidden feature-based methods were also applied. We encoded the set of high-dimensional vectors into a matrix of words and features. A matrix factorization technique was then applied to not only reduce the size of vector representation but also disclose any hidden concept information related to word meaning. The last section of Tables 3.4 and 3.5 show the results of applying

hidden features to the task of semantic distance. Although most of the correlation results slightly decreased compared to the original features, the amount dimensions of vector representation reduced significantly. In our experiment, we chose $K = 100$ as the number of singular values remaining during the process of matrix factorization. In this way, the dimension of vector presentation was also reduced dramatically to 100 while the performance results were only slightly changed and were still significantly improved compared to the baseline method.

3.7 Conclusion

In this chapter, we have presented different approaches for constructing semantic representation of words using contextual information. These approaches also provide different views on the semantic analysis of word meanings in local context. On the one hand, relation-based methods consider the relationship between words in contexts using lexical syntactic patterns, which combined with a confidence model to the determine feature representation of words. On the other hand, the local contexts are also analysed by accumulating the collocation of words within a sliding window context to disclose the word feature representation. Both methods uses the same weighting schema, PMI, for feature weighting. However, while the relation-based method utilizes a learning model to estimate the confidence value for each feature, the NGram-based method applies a different weighting filter to select appropriate features for semantic representation. The experiments show promising results from the proposed feature

generation methods compared to the related content-based methods on the task of semantic distance measurement.

Chapter 4

CONCEPTUAL TOPIC ANALYSIS FOR SEMANTIC DISTANCE

In the previous chapter, we presented an approach for semantic interpretation in which relations extracted from local contexts were used to construct meaning representation of words. This chapter introduces a new method for constructing a semantic profile of word meanings and measuring semantic distance, by using topical clues from surrounding contexts to characterise meanings of a word. With the experiment on various standard benchmarks, our method demonstrates outstanding performance compared to related methods using topical information.

4.1 Introduction

The task of measuring meaning distance between textual units is the task of finding and qualifying the strength of their semantic connections in intermediate contexts. For instance, what are the semantic clues shared by "*moon*" and "*galaxy*", or "*professor*" and "*student*", which determine how they are related? To make a judgement about meaning distance, one not only relies on what is already presented in the context, but also the amount of knowledge and experience accumulated by over larger domains.

Consequently, an appropriate approach for measuring the semantic distance be-

tween words not only utilises the connection clues from immediate contexts, but also has the ability to acquire and disclose the semantic strength of a word from background knowledge. Typical models have addressed the issue by integrating background knowledge with the task of semantic distance measurement in various ways such as explicit topic extraction (Explicit Semantic Analysis (ESA) (Gabrilovich & Markovitch 2007)), latent topic extraction (Latent Semantic Analysis (LSA)) (Landauer et al. 1998)), and salient topic extraction (Salient Semantic Analysis (SSA) (Hassan & Mihalcea 2011)). However, such methods are typical works that only focus on relationships between words and documents to induce topical information in global contexts, but ignores the the possibility that topical information from local contexts could contribute to determining word meanings as well as meaning representation. Moreover, most prior work on semantic interpretation has, up to this stage, focused only on single aspect features such as *word-based features* (Harris 1954, Turney 2001, Lin 1998*b*, Agirre et al. 2009), *topic-based features* (Gabrilovich & Markovitch 2007, Hassan & Mihalcea 2011, Radinsky et al. 2011, Deerwester et al. 1990, Dinu & Lapata 2010), with each type of feature contributing to particular aspects of meaning representation. This raises the prospect that the combination of word-based features and topical features could significantly contribute to modelling word meaning representation. To address these concerns, we propose a new method for semantic interpretation, which takes into account the immediate contexts of words to induce topical information. The topics are induced in two different ways: (1) using explicit

concepts appearing in local contexts, and (2) using word collocation in unstructured text data. Moreover, the method also examines different feature combinations which integrate word-based features and topic information for meaning representation.

By doing so, the proposed approach aims to two concerns: first, it targets modelling feature representation using topical information from local contexts. Second, it demonstrates the capacity to combine word features and topic features, as well as their performance in this combination. Details of our proposed method are presented in the sections that follow.

In the rest of this chapter, Section 4.2 presents topic analysis in local context using either unstructured text documents or Wikipedia text with Wikipedia concepts. Different combination models for meaning representation are discussed in Section 4.3. The effectiveness of the proposed method is implemented and evaluated in Sections 4.4 and 4.5 before concluding the chapter in Section 4.6.

4.2 Topical Context Analysis

This section presents two ways to approach topic analysis of text content. While the first approach aims to use explicit Wikipedia concepts to build topical representation of words, the second approach builds word representation using latent topics from unstructured text.

4.2.1 Explicit Topic Analysis

Information presented in a text document is normally considered as a mixture of multiple topics, where text content expresses topical information in different degrees. A Wikipedia article is a particular kind of text document in which the content mainly describes a particular topic called a Wikipedia concept (an explicit topic). For instance, "*Router*" is an ambiguous concept which might be interpreted in different ways. However, "*Router (computing)*" is a Wikipedia article that mainly describes a computer network device rather than a rotating cutting tool in woodworking area. Moreover, Wikipedia articles also maintain a way to connect textual content to particular topics via Wikipedia links. This mechanism enables the capture and the aggregation of topic information in a Wikipedia text (Gabrilovich & Markovitch 2007, Hassan & Mihalcea 2011). Using the idea that topical information is disclosed explicitly in a Wikipedia text, we present a method for constructing seeding information for explicit topics (Wikipedia concepts). For each Wikipedia concept, a set of words is identified as the seeds of an explicit topic. The topical clues linking these seeds is used to disclose the topical representation of words appearing in the Wikipedia repository.

4.2.2 Extracting Seeds For Wikipedia Concepts

For each Wikipedia concept, we identified a set of words that represents the meaning content of the concept. To do so, we relied on the text anchors that are linked to the concept using Wikipedia links, and the text content of the Wikipedia article itself.

Firstly, for each text anchor linked to a Wikipedia article, we extracted relationships between the text anchor and surrounding words within a sliding window context. The extraction results were then used to form a triple where the first component of the triple is the Wikipedia concept, the last component is a surrounding word and the middle component is the relationship between them. The extracted triple was thus considered as a one-dimensional representation of the concept. Processing over entire Wikipedia articles, for each Wikipedia concept, we obtained a set of triples which represented the relationship between the Wikipedia concept and its surrounding words in local context. The set of triples was then encoded into a semantic profile of the concept using a sparse vector of the surrounding words. The encoding done was very similar to the work presented in Section 3.2. Formally, given that c_i is a concept and its associated triples as $\{t^{(1)}, t^{(2)}, \ldots, t^{(n)}\}$, each $t^{(j)}$ is then encoded into the semantic space of c_i as a vector feature w_j with the constraint that $t^{(i)} = \{(c_i, r, w_j)|(w_j, r, c_i)\}$. Therefore, the sparse vector representation of the concept c_i using the Wikipedia link, $v_{concept}(c_i)$, is modelled as follows:

$$v_{concept}(c_i) = (w_1 : wt_1^{(1)}, w_2 : wt_2^{(1)}, \ldots, w_n : wt_n^{(1)}) \tag{4.1}$$

where $wt_j^{(1)}$ is the co-occurrence of the text anchor for the concept c_i and the word w_j over the entire Wikipedia text. n is the size of word dictionary on Wikipedia repository.

Secondly, as the Wikipedia concept itself is also a Wikipedia document, its sparse vector representation was modelled as a vector of words that appear in the same document. Formally, given that c_i is a Wikipedia concept, and $\{w_1, \ldots, w_n\}$ are the words in the article after removing stop-words and rare words. The sparse vector representation of the concept c_i using information from the Wikipedia document, $v_{doc}(c_i)$, is modelled as follows:

$$v_{doc}(c_i) = (w_1 : wt_1^{(2)}, w_2 : wt_2^{(2)}, \ldots, w_n : wt_n^{(2)}) \tag{4.2}$$

where $wt_j^{(2)}$ is the normalized frequency of the word w_i within the Wikipedia article c_i. n is the size of word dictionary in the Wikipedia repository.

Finally, using both information from the Wikipedia links as well as the Wikipedia articles, the sparse vector representation of the concept c_i is modelled as the combination of two different aspects $v_{concept}$ and v_{doc} as follows:

$$v_{exp}(c_i) = (w_1 : wt_1, w_2 : wt_2, \ldots, w_n : wt_n) \tag{4.3}$$

where the weighting between the word w_i and the concept c_j is measured as

$$wt_j = pmi(c_i, w_j) = \log \frac{p(c_i, w_j)}{p(c_i)p(w_j)} \tag{4.4}$$

$$p(c_i, w_j) = \frac{d(c_i, w_j)}{\sum_{i=1..m, j=1...n} d(c_i, w_j)} \tag{4.5}$$

$$d(c_i, w_j) = wt_j^{(1)} \times wt_j^{(2)} \tag{4.6}$$

$$p(w_j) = \frac{\sum_{k=1...M} d(c_k, w_j)}{\sum_{k=1..m, j=1...n} d(c_k, w_j)} \tag{4.7}$$

Equation 4.6 also shows the overall weighting of a word feature contributed by both information from Wikipedia links and Wikipedia text content.

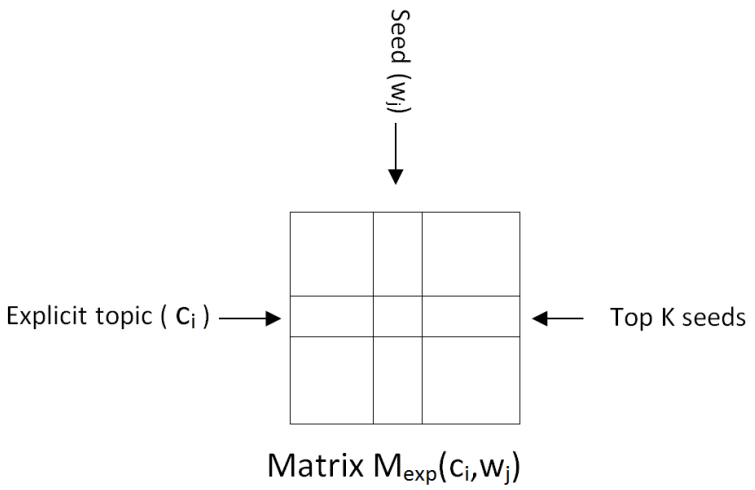


Figure 4.1: Details of the explicit topic × seed matrix $M_{exp}[topic, seed]$. Each row of the matrix is a vector of seeds, where only K-best components are selected while the rests are set to zeros

Each Wikipedia concept is represented by a sparse vector of word features. A $m \times n$ matrix M, is created by encoding all of the respective vector representations of Wikipedia concepts. Each row of the matrix represents a concept while each column is a word feature. To obtain K word features as the seeds for each Wikipedia concept, we selected K features belonging to the top K highest weighting values on each row of the matrix. The values of the rest of word features in each row were be set to zero. As a result, we obtained a reduced sparse matrix $M_{exp}[topic, seed]$ where each row was the representation of an explicit concept (topics) and each column, the representation of a seed. Figure 4.1 visualises details of the matrix.

4.2.3 Latent Topic Analysis

In the previous section, a method for extracting representation of an explicit topic using seeding words is presented. The method relies on Wikipedia text and the structure of Wikipedia links to obtain the representation. In this section, we present a different approach to obtain the seeding representation of topics using a latent analysis topic model. The method utilises large amount of plain text to infer latent topics that a word is likely to belong to. Specifically, we used Latent Dirichlet Allocation (LDA) (Blei et al. 2003) as the background topic model in building features for word representation. LDA performs a latent semantic analysis to find m latent topics in a plain text corpus.

Given a focus word w_i and a latent topic t_j, the LDA topic model produces

the probability p_{ij} that w_i belongs to a particular topic t_j. As a result, the topic representation of the word w_i is considered as a vector of latent topics, where each value of the vector represents the probability that w_i belongs to a particular topic t_j $(j = 1 \ldots m)$. The topical representation of the word w_i is described in a sparse vector $v_{lat}(w_i)$ as follows:

$$v_{lat}(w_i) = (t_1 : p_{i_1}, t_2 : p_{i_2}, \ldots, t_n : p_{i,m}) \tag{4.8}$$

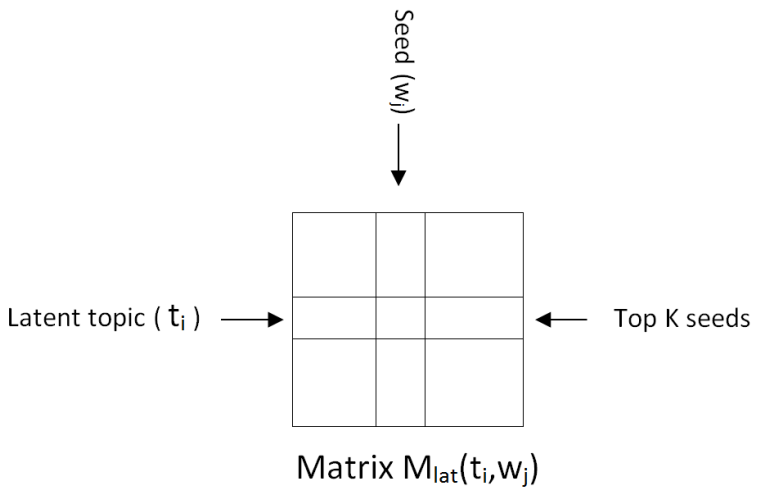


Figure 4.2: Details of the latent topic × seed matrix $M_{lat}[topic, seed]$. Each row of the matrix is a representation of a latent topics, which is constituted by a vector of seeds. Only the values of K best seeds are selected, and the rest are set to zeros.

Given a set of n words from the Wikipedia repository, the topical representation of the words is encoded in to a $m \times n$ matrix M where each row is the representation of a topic, and each column is a topic representation of a word. To obtain K words

as seeds representing each topic, we selected the top K-best values for each row and set zero for the rest of the elements in each row. As a result, we obtained a reduced matrix $M_{lat}[topic, seed]$ where each latent topic (row) only represented the maximum K seeds. Figure 4.2 visualizes the detailed matrix of latent topics and word seeds. The list of seeds from all latent topics is then used to extract topic representation of word meanings in context.

4.3 A Combination Features for Word Meaning Representation

As explained in the previous section, the meanings of a word are not only expressed using the connection clues from immediate contexts, but are also influenced by the global topics that the contexts belong to. Based on this idea, we designed an approach to construct meaning representation of words using topical information incorporated with contextual information. We designed a combination of features extracted from the local contexts as well as seeding information from global topics to construct word meaning representation. We then designed a linear combination between the set of features generated from the local contexts and the set of global topics to disclose meaning representation in context. These methods are explained in detail from the following sections.

4.3.1 Context-based Features and Topical Seeds

In the context-based analysis presented in this thesis (Chapter 3), word meanings were represented using surrounding words attached to either pattern-based relations (rel)

or within a sliding window (wnd). We denoted the vector representations constructed by such algorithms as $v_{rel}(.)$ and $v_{wnd}(.)$, respectively. By using the representations, the quality of their word features was selected by applying multiple syntactical, statistical filters. Meanwhile, words appearing in local contexts were also related to certain topics, which were determined by the topics of the surrounding words. In short, the semantic representation of a word is not only identified by the contexts where the word appears, but is also influenced by the topics linked to surrounding words.

Using this idea, this section presents a new way to combine word features in local contexts with global topical features expressed by seeding information. Given M as the topic×seed matrix, $M[topic, seed] \in \{M_{exp}[topic, seed], M_{lat}[topic, seed]\}$, and a seed-based representation of a word w_i, $v_{seed}(w_i)$, the representation of the word w_i using the combination between context-based features and seeding information is as follows:

$$v_{com}(w_i) = v_{seed}(w_i) \circ M^T[seed, topic] \tag{4.9}$$

where $v_{seed}(.)$ is a reduced vector representation from either $v_{rel}(.)$ or $v_{wnd}(.)$, which retains only those features that are considered as the word seeds of any latent/explicit topics. As the matrix $M[seed, topic]$ is induced using either explicit topic analysis (Section 4.2.1) or latent semantic analysis (Section 4.2.3), four different feature

combinations are used to construct the vector representation of $v_{com}(w_i)$ as follows:

$$v_{rel\times exp}(w_i) = exp\_seed(v_{rel}(w_i)) \circ M_{exp}^T[seed, topic] \tag{4.10}$$

$$v_{rel\times lat}(w_i) = lat\_seed(v_{rel}(w_i)) \circ M_{lat}^T[seed, topic] \tag{4.11}$$

$$v_{wnd\times exp}(w_i) = exp\_seed(v_{wnd}(w_i)) \circ M_{exp}^T[seed, topic] \tag{4.12}$$

$$v_{wnd\times lat}(w_i) = lat\_seed(v_{wnd}(w_i)) \circ M_{lat}^T[seed, topic] \tag{4.13}$$

where $exp\_seed(.)$ is a function which retains only features as seeds of latent topics from the input vector, and $lat\_seed(.)$ is also a function which retains only features as seeds of latent topics from the input vector.

4.3.2 Context-based Features and Topical Features

First, using local context analysis, meaning representation of a word w_i is constructed as either a vector of relational features, $v_{rel}(w_i)$ or a vector of NGram word features, $v_{wnd}(w_i)$. These representative vectors are rewritten as dense vectors from Equations 3.13 and 3.17 as follows:

$$v_{rel}(w_i) = (w_1, w_2, \ldots, w_{n_{rel}}) \tag{4.14}$$

$$v_{wnd}(w_i) = (w_1, w_2, \ldots, w_{n_{wnd}}) \tag{4.15}$$

where n_{rel} and n_{wnd} are the number of relational features and NGram word features, respectively.

Second, using explicit topic analysis (see Section 4.2.1), the word w_i is represented by a vector of explicit topical features, $v_{exp}(w_i)$, which is described as a dense vector as follows:

$$v_{exp}(w_i) = (t_1, t_2, \ldots, t_{m_{exp}}) \tag{4.16}$$

where m_{exp} is the number of explicit features.

Similarly, using latent topic analysis (see Section 4.2.3), the word w_i is represented by a vector of latent topical features, $v_{lat}(w_i)$, which is described as a dense vector as follows:

$$v_{lat}(w_i) = (t_1, t_2, \ldots, t_{m_{lat}}) \tag{4.17}$$

where m_{lat} is the number of latent features.

Given a word w_i appearing in context, its semantic representation is described either using relational vectors $v_{rel}(w_i)$ or NGram-based vector $v_{wnd}(w_i)$. Meanwhile, the topic-based representation of the word w_i is expressed using either a vector of explicit topics $v_{exp}(w_i)$ or a vector of latent topics $v_{lat}(w_i)$. There are four possibilities that features from local contexts and topical information can be combined.

First, the following equation presents the linear combination between relational

features in local contexts and explicit topic features.

$$\begin{aligned} v_{rel+exp}(w_i) &= (\alpha^{(1)} v_{rel}(w_i), \beta^{(1)} v_{exp}(w_i)) \\ &= (\alpha^{(1)} w_1, \ldots, \alpha^{(1)} w_{n_{rel}}, \beta^{(1)} t_1, \ldots, \beta^{(1)} t_{m_{exp}}) \end{aligned} \tag{4.18}$$

We used $\alpha^{(1)}$ and $\beta^{(1)}$ as the trade-off parameters to adjust the contribution of each respective representation. These parameters were then independently evaluated during the experiment.

Second, a representation of the word w_i was constructed as the combination between relational features and latent topics as follows:

$$\begin{aligned} v_{rel+lat}(w_i) &= (\alpha^{(2)} v_{rel}(w_i), \beta^{(2)} v_{lat}(w_i)) \\ &= (\alpha^{(2)} w_1, \ldots, \alpha^{(2)} w_{n_{rel}}, \beta^{(2)} t_1, \ldots, \beta^{(2)} t_{m_{lat}}) \end{aligned} \tag{4.19}$$

where $\alpha^{(2)}$ and $\beta^{(2)}$ are the trade-off parameters.

Next, when constructing word representation using the combination between NGram word features in a sliding window context and topical features, the representation of the word w_i is as follows:

$$\begin{aligned} v_{wnd+exp}(w_i) &= (\alpha^{(3)} v_{wnd}(w_i), \beta^{(3)} v_{exp}(w_i)) \\ &= (\alpha^{(3)} w_3, \ldots, \alpha^{(3)} w_{n_{wnd}}, \beta^{(3)} t_1, \ldots, \beta^{(3)} t_{m_{exp}}) \end{aligned} \tag{4.20}$$

where $\alpha^{(3)}$ and $\beta^{(3)}$ are the trade-off parameters.

Final, the representation of the word w_i using the combination of NGram word features and latent topic features can be described as follows:

$$\begin{aligned} v_{wnd+lat}(w_i) &= (\alpha^{(4))} v_{wnd}(w_i), \beta^{(4)} v_{exp}(w_i)) \\ &= (\alpha^{(4)} w_4, \ldots, \alpha^{(4)} w_{n_{wnd}}, \beta^{(4)} t_1, \ldots, \beta^{(4)} t_{m_{lat}}) \end{aligned} \tag{4.21}$$

where $\alpha^{(4)}$ and $\beta^{(4)}$ are the trade-off parameters.

4.3.3 Evaluation of Feature Combinations

In summary, given a word w_i, its meaning representation can be constructed using one of eight feature vectors as follows:

$$v_{rel+exp}(w_i); v_{rel+lat}(w_i); v_{wnd+exp}(w_i); v_{wnd+lat}(w_i)$$

$$v_{rel\times exp}(w_i); v_{rel\times lat}(w_i); v_{wnd\times exp}(w_i); v_{wnd\times lat}(w_i)$$

The effectiveness of each feature combination was evaluated using the task of semantic distance measurement. Given two words w_i and w_j and their respective vector representations $v_{com}(w_i)$ and $v_{com}(w_j)$, the semantic distance of w_i and w_j was estimated by using standard Cosine distance between their combination vectors as

follows:

$$dist(w_i, w_j) = \frac{v_{com}(w_i) \times v_{com}(w_j)}{\|v_{com}(w_i)\| \times \|v_{com}(w_j)\|} \tag{4.22}$$

Table 4.1: Experiment on MTruk for tuning parameters. Four different feature combinations that require parameter turning were experimented. For each feature combination, we selected values of parameters that returned the best correlation results during the semantic distance measurements

Algorithms	Parameters	$\rho \times 100$
Explicit Semantic Analysis (Radinsky et al. 2011)	–	59
Temporal Semantic Analysis (Radinsky et al. 2011)	–	63
Relational Features + Explicit Topic Features ($v_{rel+exp}$)	$\frac{\alpha^{(1)}}{\beta^{(1)}} = 0.002$	62.4
Relational Features + Latent Topic Features($v_{rel+exp}$)	$\frac{\alpha^{(2)}}{\beta^{(2)}} = 0.018$	61.9
NGram-based Features + Explicit Topic Features($v_{wnd+exp}$)	$\frac{\alpha^{(3)}}{\beta^{(3)}} = 0.002$	64.7
NGram-based Features + Latent Topic Features($v_{wnd+exp}$)	$\frac{\alpha^{(4)}}{\beta^{(4)}} = 0.016$	63.6

4.4 Experiments

4.4.1 Testing Benchmarks

We used two popular semantic distance benchmarks, WS-353 and RG-65, to evaluate the effectiveness of the proposed features for semantic representation. We also used MTruk testing benchmarks acting as independent tests for scanning trade-off parameters (see Section 2.3.1 for more detail about these datasets).

4.4.2 Text Corpus

To be consistent with the experiments from the previous chapter, we also used the first $1,000,000$ Wikipedia articles from the English XML dump of October 01, 2012 for extracting word representation in contexts (see Section 3.5.2). Moreover, these articles were also used to extract seeding information as well as explicit topics.

To extract latent topic features, we used plain texts from the first $100,000$ Wikipedia documents to feed to the LDA training model. The reasons for us to choose this smaller amount of documents is because the LDA training phrase was time consuming when processing a large amount of documents. We expected to reduce the number of input documents but kept the relatively large word dictionary to cover most of the expected words.

4.4.3 Extracting latent topics and seeding representatives

To extract latent topics and seeds for each topic, stop-words from 100,000 Wikipedia articles was removed. We also applied a stemming technique to transform words in various families into root words. Rare words were also removed by using document frequency threshold ($df = 5$). After the pre-processing step, the articles were used to train a latent topic model using LDA training model, we used GibbsLDA++ Phan et al. (2008) implementation with the standard configuration. We also selected $m = 1,000$ as the number of expected latent topics. As a result, a matrix of latent topics and words was obtained. We also received $190,132$ word features in totally. Each

element of the matrix could be interpreted as the probability that each word belonged to a topic. Each column was a vector of latent topics which represented the semantic meanings of the respective words using latent topics features. These vectors were evaluated on the task of measuring semantic distance between words using only latent topics as features.

To select the number of seeds for each topic, we selected top $K = 20$ best values for each topic and the respective word features becomes seeds of the topics. The remaining value for each topic will be set to zeros. The matrix M became a sparse matrix where each column was the representation of a word seed using latent topic features. These seeds then contributed to the word meaning representation using local context information and topical information.

4.4.4 Extracting explicit topics and seeding representatives

The $1,000,000$ Wikipedia articles have been used to extract word representation using explicit topics. We first applied the relation extraction technique described in Section 4.2.1 to extract word-concept associations over the local contexts and Wikipedia document context. Specifically, we limited the selection to only concepts that had more than 100 associations with surrounding words over the entire Wikipedia corpus. After assigning the weights for word-concept associations, each concept was also considered as an explicit topic and was represented by a vector of word features. These vectors were later evaluated on a semantic distance measurement between words us-

ing only explicit topics as features. To select seeds for each explicit topic, we also selected top $K = 20$ best word features as seeds for each explicit topic and the rest of vector values set to zeros. The explicit topic was represented as a set of K seeds. The collection of all of the vectors produced a sparse concept-word matrix, where each column of the matrix was represented for a seed and was also a vector of explicit topics.

Secondly, instead of using patterns to extract word-concept associations, we used NGram-based technique to extract any word-concept associations within a sliding window (ws=3) (see Section 3.3.1) with the procedures being organized using exactly the same pattern-based technique for weighting and selecting top K seeds. As a result, the semantic representation of a word using explicit topics was also presented, which was then tested through the task of semantic distance measurement. Additionally, the collection of the vectors returned a topic-seed matrix, which was then used for feature combination.

4.4.5 Choosing Parameters for Feature Combination

Section 4.3.2 has presented the linear combinations between word features and topical features. As both sets of word features and topical features were constructed on different weighting schemas, changing the parameters $\alpha^{(*)}$ and $\beta^{(*)}$ would affect the influences of those sets of features. In particular, four different pairs of trade-off parameters $\{< \alpha^{(1)}, \beta^{(1)} >, < \alpha^{(2)}, \beta^{(2)} >, < \alpha^{(3)}, \beta^{(3)} >, < \alpha^{(4)}, \beta^{(4)} >\}$ need to be

estimated. We implemented these feature combination methods, {relational features + explicit topic features}, {relational features + latent topic features}, {NGram word features + explicit topic features}, {NGram word features + latent topic features}, on the independent semantic distance MTruk benchmark. There was no single pair of parameters that returned the best result for all the methods. We therefore selected a pair of parameters that returns the best correlation for each feature generation on the MTruk dataset. The correlation results and the parameters for each method are shown in Table 4.1. The selected values of these parameters were used to construct vector representations of words, which were then evaluated using the WS-353 and RG-65 benchmarks.

4.5 Evaluation

In this section, we discuss the effectiveness of the proposed topical-based features as well as their combinations with features extracted from local contexts. Four types of topic-based features can be grouped together: (1) latent topic features, (2) explicit topic features, (3) linear combination features, and (4) seed-based combination features.

4.5.1 Overall results compared to other topic-based methods

Tables 4.2 and 4.3 show the overall results for all the proposed feature generations compared to related methods on the task of semantic distance measurement. All of the proposed features generated from our method show a promising correlation with

Table 4.2: Comparison results with different content-based methods on WS-353 datasets using Spearman's rank correlation (ρ). † indicates the results using the selected parameters from the independent dataset

Algorithms	$\rho \times 100$
Latent Topic Features (LSA) (Finkelstein et al. 2001)	58.10
Latent Topic Features (LDA) (Dinu & Lapata 2010)	53.39
Explicit Semantic Analysis (ESA) (Gabrilovich & Markovitch 2007)	74.8
Salient Semantic Analysis (SSA) (Hassan & Mihalcea 2011)	67.0
Latent Topic Features (v_{lat})	67.01
Explicit Topic Features (v_{exp})	69.35
Relational Features + Explicit Topic Features ($v_{rel+exp}$)	**74.2**†
Relational Features + Latent Topic Features($v_{rel+exp}$)	73.6†
NGram-based Features + Explicit Topic Features($v_{wnd+exp}$)	73.9†
NGram-based Features + Latent Topic Features($v_{wnd+exp}$)	73.52†
Relational Features $\times$ Explicit Topic Features ($v_{rel\times exp}$)	72.48
Relational Features $\times$ Latent Topic Features($v_{rel\times lat}$)	71.92
NGram-based Features $\times$ Explicit Topic Features($v_{wnd\times exp}$)	70.64
NGram-based Features $\times$ Latent Topic Features($v_{wnd\times lat}$)	69.81

human judgements compared to the baseline method, LSA, and comparable results to other methods based on topical information.

First, the results from the latent topic generation indicate the advantages of the method when modelling meaning representation of words. With certain limitations on the amount of features used (1000 latent topics), the method shows improved results compared to similar implementations such as LSA and LDA (58.10, 53.39% vs. 67.01% on WS-353, and 60.9% vs. 63.93% on RG-65). A possible explanation could be that there was a reduction of noise from input data as well as better coverage and diverse of the Wikipedia text.

Second, when taking the advantages of Wikipedia concepts as explicit topics as

well as their relations with words in Wikipedia text, the explicit topic feature generation shows significant improvement in its correlation with human judgements compared to the baseline method (58.10% vs 69.35% on WS-353). Compared to methods using explicit topics such as ESA and SSA, our proposed features return comparable results on the RG-65 corpus (74.9%, 83.3% vs. 76.15%). This also indicates the effectiveness of background knowledge on semantic interpretation and semantic distance measurement.

Next, when linear concatenating either of these topical features with the features generated from local contexts, the results are significantly improved for each of the combinations. Especially, the independent trade-off parameters, the best results were achieved as 74.12% on the WS-353 and 86.17% on the RS-65. This indicates the contribution of features from multiple semantic aspects on meaning representation as well as on semantic distance measurement.

Finally, instead of using all of the word features for feature combinations, only using features such as topical seeds showed the effectiveness on semantic distance measurement, which was also illustrated by the correlation results from the last section of Tables 4.2 and 4.3. Although the results from seed-based features were not as high as those from the linear combination, the results were still outstanding to most of the methods using a single feature aspect. Specially, this was the case when a significant amount of non-seed features were reduced. Thus, the proposed methods indicate their usefulness in semantic analysis as well as semantic distance measurement.

Table 4.3: Comparison results with different content-based methods on RG-65 datasets using Spearman's rank correlation (ρ).

Algorithm	$\rho \times 100$
Latent Topic Features (LSA) (Finkelstein et al. 2001)	60.9
Explicit Semantic Analysis (ESA) (Hassan & Mihalcea 2011)	74.9
Salient Semantic Analysis (SSA) (Hassan & Mihalcea 2011)	83.3
Latent Topic Features (v_{lat})	63.93
Explicit Topic Features (v_{exp})	76.15
Relational Features + Explicit Topic Features ($v_{rel+exp}$)	**86.17**
Relational Features + Latent Topic Features($v_{rel+lat}$)	84.62
NGram-based Features + Explicit Topic Features($v_{wnd+exp}$)	81.15
NGram-based Features + Latent Topic Features($v_{wnd+lat}$)	78.82
Relational Features $\times$ Explicit Topic Features ($v_{rel\times exp}$)	84.65
Relational Features $\times$ Latent Topic Features($v_{rel\times lat}$)	82.17
NGram-based Features $\times$ Explicit Topic Features($v_{wnd\times exp}$)	79.66
NGram-based Features $\times$ Latent Topic Features($v_{wnd\times lat}$)	76.95

4.5.2 Latent Topic Features vs. Explicit Topic Features

In comparing latent topic features with explicit topic features for feature generation, we start with their fundamental ideas. In the first instance, latent topic generation brings benefits because it can operate on different unstructured text domains and is beneficial to content-based methods. Compared to the explicit topic generation, we relied on the characteristics of Wikipedia articles and the Wikipedia links to identify word-concept relations. Moreover, with explicit topic features, high dimensional feature vector is one of the issues to deal with. However, the simplicity of implementation is also a plus point when adopting this feature generation method.

In terms of performance, the results from Tables 4.2 and 4.3 show that the explicit

topic features return slightly improved results compared to latent topic features in the ways that they were used either independent or combined with other feature aspects. However, the trade-off can be seen clearly via their high dimensional vectors even if when combined with seeding information.

4.5.3 Linear Combination vs. Seed-based Combination

Linear feature combination is a straight-forward way to integrate features from different aspects. However, its mechanism can be also affected by the trade-off parameters α and β. Meanwhile, seed-based feature combinations mainly operate by combining selected features as seeds and with topical information. With feature combinations, the number of dimensions in a seed-based combination was significantly reduced compared to a linear one. This is also a trade-off in correlation results over four different ways of combining features.

4.6 Conclusion

In this chapter, we have presented a set of topical-based methods to construct semantic representation of words as well as a means to measure semantic distance. The methods take into account the combinations between words meanings in the local contexts with topical information. The methods include different techniques for generating topical features and their combinations. In the experiments on different datasets, the results indicated the effectiveness of multiple feature aspects on semantic interpretation and on semantic distance measurement.

Chapter 5

MULTI-WAY FEATURE ANALYSIS FOR SEMANTIC INTERPRETATION

We have presented two aspects of feature analysis for meaning representation in the previous chapters. In this chapter, we address the third aspect for feature analysis which utilises word meanings in context as well as structural information.

This chapter proposes and evaluates a tensor-based method for semantic interpretation using a content-based model. Such a model that builds the representation of words directly from text and does not require pre-existing linguistic knowledge. While traditional vector-based models have been used for different tasks in representing word meaning, drawbacks have occurred because the model do not incorporate sufficient structural information such as word order, and syntactic information. Taking into account recent work to overcome this weakness, we propose a new method that utilises tensor analysis to build representation of word meaning. Our content-based models demonstrate significantly improved performance when compared to a robust baseline model on a number of semantic distance measures.

5.1 Introduction

In the tasks of semantic processing, many approaches have argued that word meanings can be modelled by considering their distributions within a given context (Schutze 1993). Word meanings are represented by collecting word collocation frequencies and encoding them into a high-dimensional *context vector* (Turney et al. 2010). In a number of research works, the word meaning representation has shown acceptable performance in a number of cognitive tasks (Jones & Mewhort 2007, Landauer & Dumais 1997).

However, there has been criticism that a *context vector* representation does not incorporate structural information from a text such as word order or syntactical aspects (Turney et al. 2010, Symonds et al. 2012). For instance, in the task of understanding word meanings, roles of syntax and word order are significant. According to De Saussure (1916/1996), two fundamental word associations are believed to disclose word meaning: *syntagmatic* and *paradigmatic* associations. They heavily influence the way humans reason word meanings and directly use word order and syntactic information as the basic structures for meaning manipulation. This emphasizes the importance of structural information in processing word meanings as well as in semantic processing in general. However, the fact that the context vector representation ignores such information raises the question whether an alternative representation could maintain both local context features as well as structural information in mean-

ing understanding.

In this chapter, we propose a new content-based method to represent word meanings in a way that maintains not only local context information as a context vector, but also structural information such as word order and syntactical aspects. The content-based model presented captures words in context as triples of lexical syntactic structure. These are then encoded into three-way tensor, which helps to maintain both word associations and word order. In this stage, the representation of words can be achieved using characteristics of tensor analysis. Furthermore, the existing encoding tensor can be further processed using tensor decomposition techniques to disclose any latent factors representing word meaning. This step presents a second model of tensor analysis on word meaning. The model is then evaluated using various measures on semantic distance. The correlation results with human judgements show significant improvement on the first model and comparable on the second model when compared to the popular baseline method using the same benchmarks.

In the rest of this chapter, Section 5.2 presents related work on semantic interpretation using content-based models as well as the basic understandings to support for the continuing sections. Tensor analysis on contextual information is presented in Section 5.3, which includes memory tensor analysis model (Section 5.3.2) and latent tensor analysis model (Section 5.3.3). Implementation and evaluation are then addressed in Section 5.4 and 5.5 before concluding the chapter in Section 5.6.

5.2 Related Work

The main areas of research that provide a theoretical background for our models include: (1) the distribution hypothesis defining word meaning, (2) the use of semantic space to define word meaning, and (3) tensor analysis to capture structural information. These are discussed below.

5.2.1 Word Meanings in Distributional Contexts

Researchers have argued that meanings of a word can be modelled by considering its distributions in a text. In doing so, they rely fundamentally on a set of assumptions about the nature of language and meaning, which is referred to as *the distributional hypothesis*. The hypothesis has been described in different ways: for example as "words which are similar in meanings occur in similar contexts" (Rubenstein & Goodenough 1965), "words with similar meanings will occur with similar neighbours if enough text material is available" (Schutze & Pedersen 1995), and as "a representation that captures much of how words are used in natural context will capture much of what we mean by meaning" (Landauer & Dumais 1997). Inspired by the distributional hypothesis, many research works have been proposed to model word meanings (Landauer & Dumais 1997, Lund & Burgess 1996, Lin 1998*a*, Turney et al. 2010).

In initial research in the field, for example, De Saussure (1916/1996) argued that the meanings of a word can be revealed from the relationships between words. The

author emphasized two types of association between words that creates meanings: *syntagmatic* and *paradigmatic* associations. A syntagmatic relation is said to take place between two words if they co-occur more frequently than expected. The word pairs *coffee:drink*, *sun:hot*, or *teacher:school* are typical examples of a syntagmatic relation as they frequently co-occur in texts. On the other hand, two words have a paradigmatic relation if they are able to substitute for one another in sentences without changing the grammaticality or acceptability of the sentences. Word pairs that are synonyms or antonyms like *quick:fast*, or *eat:drink* are typical examples of a paradigmatic relation. Normally, words in having a paradigmatic relation are the same form of part of speech, whereas words in a syntagmatic relation can but need not be the same part of speech (Rapp 2002). Ideas based on syntagmatic and paradigmatic relations have motivated other works on the semantic space of words such as that by Sahlgren et al. (2008). Turney (2006) also inherited the idea to define relational and attributional similarity between words.

Different from previous approaches to word meaning representation that take into account any word-word associations, we consider the triples of *entity-relation-entity* type in a text while still maintaining word order for other tasks that utilises syntagmatic and paradigmatic associations.

5.2.2 Creating the Word Semantic Space

Popular approaches to the distributional hypothesis aim to collect word co-occurrences and frequencies and encode them into a high-dimensional *context vector*, which is also referred to as the semantic space of the word (Turney et al. 2010). One of the most well-known semantic space models of words is the Hyperspace Analogue to Language (HAL) (Lund & Burgess 1996). HAL builds context vectors by storing word co-occurrence frequencies in a word-by-word matrix. However, as the number of words in the vocabulary grows, the HAL matrix becomes very large. Therefore, it is common to only use the co-occurrence frequencies of the top k most frequent words in the corpus to represent word meaning. Inspired by the ideas of HAL, many recent approaches have proposed working with larger amounts of data on the Web. Agirre et al. (2009), for example, attempted to extend the HAL model using different sliding windows on Web data. In contrast to HAL-based models, Latent Semantic Analysis (LSA) considers word-document co-location to build a context matrix. The mathematical technique, known as single value decomposition (SVD) (Golub & Van Loan 1989), has also been used to reduce the dimensions of the context matrix to the k most significant latent concepts (Landauer et al. 1998). However, while models based on LSA and HAL have been shown to simulate human performance on a number of cognitive tasks, it has been argued that these models are limited in capturing structural information such as word order, and syntactical information or achieving other basic cognitive language abilities (Perfetti 1998, Jones & Mewhort 2007). More

recently, a number of semantic space models have attempted to increase the amount of structural information encoded within a representation (Jones & Mewhort 2007, Sahlgren et al. 2008). In particular, Symonds et al. (2012) have proposed an efficient encoding mechanism to builds on word space models by adding order information. This method is similar to our proposal, but instead of using a *memory matrix*, we encode contextual information into a three-way tensor and utilise tensor analysis techniques to disclose the meaning representation of words.

5.2.3 Tensor Analysis

Multi-way tensors have been used to construct different kinds of word space models in recent years. Turney (2007), for example, used a word-word-pattern tensor to model semantic similarity. Van de Cruys et al. (2013) proposed a method to induce a latent model for word composition, while Baroni & Lenci (2010) proposed a general, tensor-based framework for structured word space models.

Tensor analysis has been used in our work in two different ways. We first utilised the three-way model of tensor to encode information from triples. The benefits of this organization was that it maintained both word associations in context as well as word order structural information. Secondly, by using the natural structure of tensor, we were able to compute tensor decomposition to disclose latent features for the meaning representation of words. Both of these uses were then evaluated using the task of measuring semantic distance between words.

5.3 Tensor Analysis for Meaning Representation

5.3.1 Multi-way Arrays

Multi-way arrays, often referred to as tensors, are higher-order generalizations of vectors and matrices (Acar & Yener 2009). A high-order array is represented as $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \ldots \times I_N}$, where the order of $\mathcal{X}$ is $N (N > 2)$, and the number of dimensions while a vector and a matrix is an array of orders 1 and 2, respectively. Its terminology is also different compared to that of vectors and matrices.

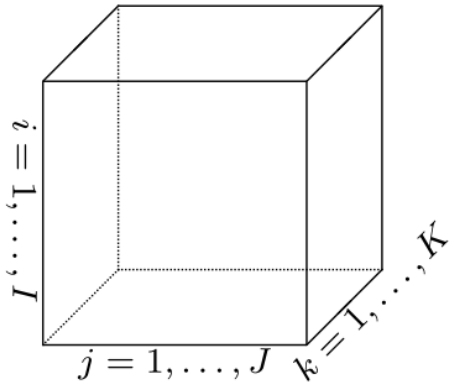


Figure 5.1: The cube represents a model of a third-order (three-way) tensor: $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$. I, J, and K are the number of dimensions for each mode (way) (Kolda & Bader 2009)

Each dimension of a tensor is called *a mode* (a way) and the number of variables in each mode is used to indicate the dimensionality of a mode. For instance, $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ is a multi-way array with 3 modes (called a three-way order tensor). I, J, and K are dimensions in the first, second and third modes, respectively. Each entry of $\mathcal{X}$ is denoted as x_{ijk}, which is in the i^{th} row, j^{th} column and k^{th} tube of $\mathcal{X}$. Figure 5.1

visualizes the third-order tensor: $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ (Kolda & Bader 2009).

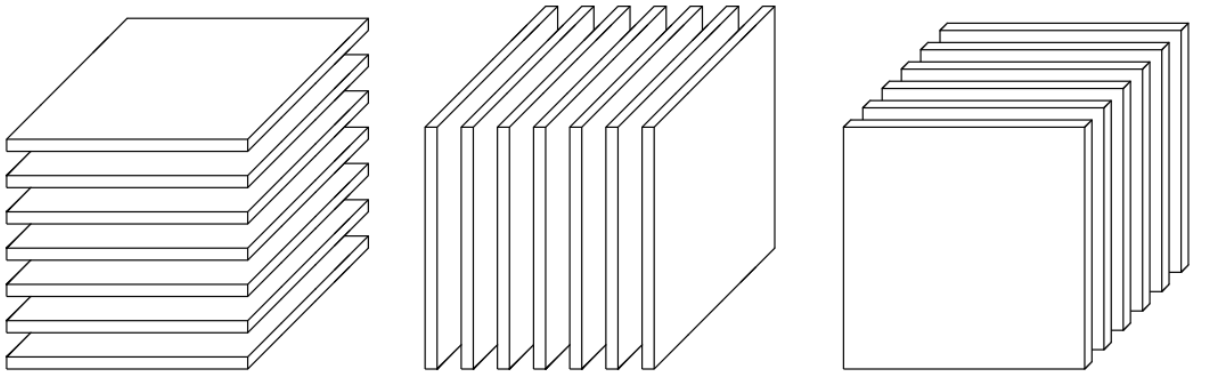
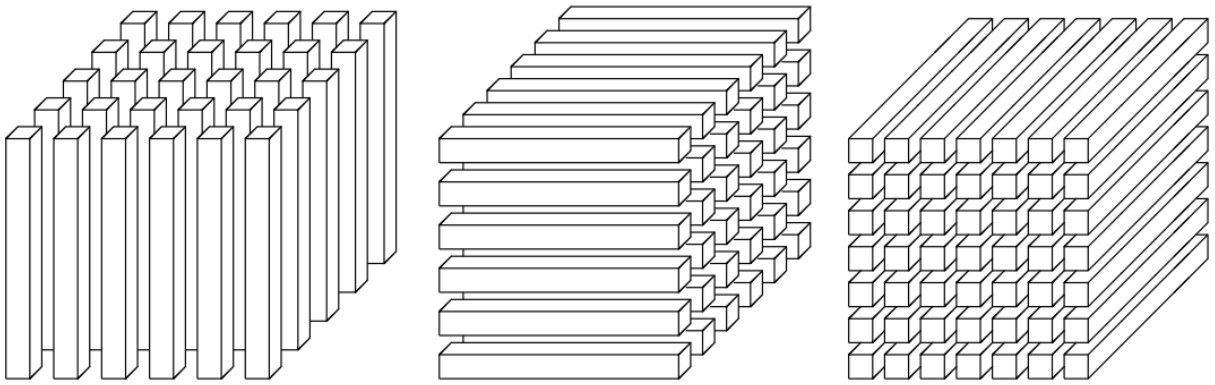


Figure 5.2: Slices of a third-order tensor. Horizontal slices, lateral, frontal slices are obtained by fixing the first, second, and third model of the tensor respectively.

When a subset of tensor indices are fixed, sub-arrays are formed. Slices are two-dimensional sections of a tensor, defined by fixing all but two indices. For instance, in a three-way tensor, horizontal, lateral, and frontal slides are denoted as $\mathcal{X}_{i::}$, $\mathcal{X}_{:j:}$, and $\mathcal{X}_{::k}$, which can be created by fixing the first, second and third modes, respectively. Figure 5.2 depicts such tensor slices (Kolda & Bader 2009).

Similarly, fibers of a tensor are identified by fixing every index but one. In a third-order tensor, columns are mode-1 fibers denoted as $\mathcal{X}_{:jk}$, rows are mode-2 fibers denoted as $\mathcal{X}_{i:k}$, and tubes are mode-3 fibers denoted as $\mathcal{X}_{ij:}$. When extracted from the tensor, the fibers are always assumed to be oriented as column vectors. Figure 5.3 shows the fibers as columns, rows and tubes respectively.



(a) Mode-1 (column) fibers: $\mathbf{x}_{:jk}$ (b) Mode-2 (row) fibers: $\mathbf{x}_{i:k}$ (c) Mode-3 (tube) fibers: $\mathbf{x}_{ij:}$

Figure 5.3: Fibers of a third-order tensor. Column, Raw, or Fiber columns are obtained by varying either the first, second, or the third mode of the tensor respectively and fixing the rest remaining modes (Kolda & Bader 2009)

5.3.2 Modelling the Semantic Space using a Three-way Tensor

Most research works on modelling the semantic space from the literature used vector or matrix to encode words in context. Typical work is that of HAL (Lund & Burgess 1996) and LSA (Landauer et al. 1998), which utilised word pairs co-located in the text surface as well as their frequency to construct semantic space of words (Turney et al. 2010). Tensors, on the other hand, are able to model multilateral entities in context based on their natural structure. In this section, we describe how to encode contextual information into a three-way tensor that offers facilities for analysing semantics.

Consider the contextual sentence: "*Garlic can also kill harmful bacteria, fungi, and viruses*". The sentence is fed into a relation extraction model and a list of extracted triples is returned such as T_1=(*garlic*, *kill*, *bacteria*), T_2=(*garlic*, *kill*, *fungi*), T_3=(*garlic*, *kill*, *virus*). Before encoding the triples into three-way tensor, terms from

Table 5.1: An example of indexed and initialized terms from the sentence "*Garlic can also kill harmful bacteria, fungi, and viruses*"

Term-id	Term	Initalized Vector
1	garlic	$I_{garlic} = (1\ 0\ 0\ 0\ 0)^T$
2	kill	$I_{kill} = (0\ 1\ 0\ 0\ 0)^T$
3	bacteria	$I_{bacteria} = (0\ 0\ 1\ 0\ 0)^T$
4	fungi	$I_{fungi} = (0\ 0\ 0\ 1\ 0)^T$
5	virus	$I_{virus} = (0\ 0\ 0\ 0\ 1)^T$

the triples need to be indexed and initialized. Table 5.1 shows the process by which the terms are indexed and initialized. In the process, each term that is indexed generates an initialized vector with zero values except for value 1 in its indexed position. The vector represented for each term is then combined to encode the triples into a memory tensor.

Memory Tensor

A memory tensor is used to encode extracted triples. Given an extracted triple $t_{ijk} = (e_i, r_j, e_k)$ and $I_{e_i}, I_{r_j}, I_{e_k}$ as the initialized vectors of the triple components, respectively, a tensor slice, when fixing r_j, is constructed as follows:

$$\mathcal{X}_{:r_j:} = I_i \otimes I_k^\top \tag{5.1}$$

where $\otimes$ is the Kronecker product of the initialized vectors. The resulting $\mathcal{X}_{:r_j:}$ becomes a lateral slice (matrix) with a structure where e_i is a row and e_k is a column.

The orders of the triple becomes the order in a tensor with each component of the tensor being considered as a mode in a three-way tensor. Similarly, when fixing e_i and e_k, the respective tensor slices $\mathcal{X}_{e_i::}$ $\mathcal{X}_{::e_k}$ are obtained as:

$$\mathcal{X}_{e_i::} = I_j \otimes I_k^\top \tag{5.2}$$

$$\mathcal{X}_{::e_k} = I_i \otimes I_j^\top \tag{5.3}$$

Consider an example of encoding only a single triple (*garlic*, *kill*, *bacteria*) into tensor slices when fixing one of the components of the triple. The rows and columns of the tensor slice express the interaction between the two other components of the triple.

$$\mathcal{X}_{garlic,:,:} = I_{kill} \otimes I_{bacteria}^\top = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{5.4}$$

$$\mathcal{X}_{:,kill,:} = I_{garlic} \otimes I_{bacteria}^\top = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{5.5}$$

$$\mathcal{X}_{:,:,bacteria} = I_{garlic} \otimes I_{kill}^{\top} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (5.6)$$

Once multiple triples that share the same triple component such as r_j are extracted, the tensor slide is measured as the sum of all possible slices associated with the component as follows:

$$\mathcal{X}_{:r_j:} = \sum_{e_i,e_k|\exists(e_i,r_j,e_k)} I_i \otimes I_k^{\top} \quad (5.7)$$

where I_{e_i} and I_{e_k} are initialized vectors of each possible pairs e_i and e_k respectively. Similarly, the aggregate slices of e_i and e_k are also constructed as follows:

$$\mathcal{X}_{e_i::} = \sum_{r_j,e_k|\exists(e_i,r_j,e_k)} I_j \otimes I_k^{\top} \quad (5.8)$$

$$\mathcal{X}_{::e_k} = \sum_{e_i,r_j|\exists(e_i,r_j,e_k)} I_i \otimes I_j^{\top} \quad (5.9)$$

Consider the triples T_1,T_2,T_3 extracted from the sentence "Garlic can also kill harmful bacteria, fungi, and viruses". The following formulas show the aggregated slices for

each triple component as showed in Figure 5.4.

$$\mathcal{X}_{garlic,:,:} = I_{kill} \otimes I_{bacteria}^{\top} + I_{kill} \otimes I_{fungi}^{\top} + I_{kill} \otimes I_{virus}^{\top} \tag{5.10}$$

$$= \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathcal{X}_{:,kill,:} = I_{garlic} \otimes I_{bacteria}^{\top} + I_{garlic} \otimes I_{fungi}^{\top} + I_{garlic} \otimes I_{virus}^{\top} \tag{5.11}$$

$$= \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The result of encoding the triples associated with a word is a tensor slice. Depending on the position of the word, different tensor slices could be used to express the word meanings in context. For instance, the word "*garlic*" is possible to place as either the first or the third component of a triple as it can act as either a subject or an object of a sentence. In that case, a horizontal slice and a frontal slice are used to model the word "garlic" in a tensor structure.

We have presented how to encode triples into a three-way tensor structure by breaking the tensor structure into tensor slices. However, it is more complicated when constructing a tensor by combining slices from different modes. Instead, the tensor is created by accumulating one triple at a time. Given a triple $T_n = (e_i, r_j, e_k)$ and $I_{e_i}, I_{r_j}, I_{e_k}$ as initialized vectors of each triple component, the following equation shows how to encode the triple T_n into a tensor.

$$\mathcal{X}_{T_n} = \left(I_{e_i} \otimes I_{r_j}^{\top}\right) \otimes I_{e_k}^{\top} \tag{5.12}$$

The Kronecker product between the three vectors creates a three-way tensor, whose modes are represented components of the triple, respectively. Given a set of N triples, a three-way tensor is constructed by encoding the triple set as follows:

$$\mathcal{X} = \sum_{n=1}^{N} \mathcal{X}_{T_n} = \sum_{T_n=(e_i,r_j,e_k),n=1\ldots N} \left(I_{e_i} \otimes I_{r_j}^{\top}\right) \otimes I_{e_k}^{\top} \tag{5.13}$$

For instance, given a set of triples $\{T_1, T_2, T_3\}$, the following equation shows how a

tensor $\mathcal{X}$ is constructed, as shown in Figure 5.4.

$$\begin{aligned}\mathcal{X} &= \left(I_{garlic} \otimes I_{kill}^{\top}\right) \otimes I_{bacteria}^{\top} + \left(I_{garlic} \otimes I_{kill}^{\top}\right) \otimes I_{fungi}^{\top} + \left(I_{garlic} \otimes I_{kill}^{\top}\right) \otimes I_{virus}^{\top} \\ &= \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \\ &+ \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}\end{aligned} \tag{5.14}$$

Figure 5.4 shows the result after encoding all three T1, T2, and T3 triples into a three-way tensor. The tensor slices are also extracted after fixing a respective component of the triples. With this organization, the structure and order information from the triples are maintained, and also facilitates further semantic processing tasks.

Capturing the Closeness of Triples

The memory tensor is used to manage the encoding information of triples. However, the closeness of triple components is different for each triple. To catch the closeness

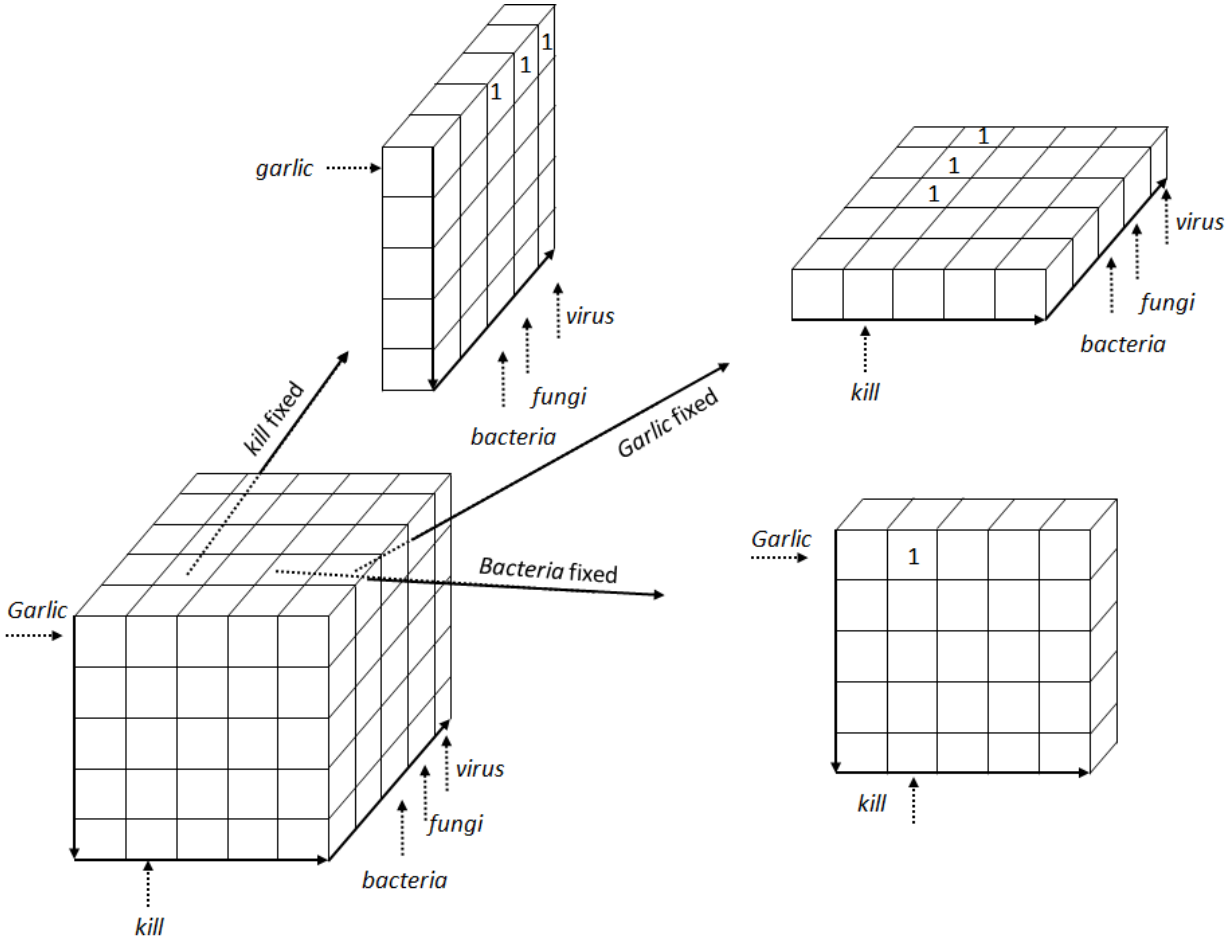


Figure 5.4: A three-way tensor as a result of constructing three triples T_1=(*garlic*, *kill*, *bacteria*), T_2=(*garlic*, *kill*, *fungi*), T_3=(*garlic*, *kill*, *virus*). The respective slice for each mode is also depicted when fixing the respective mode, which shows how the encoding values are assigned

of components within each triple, we used the text distance between these components as the weighting adjustment for each triple. As a result, the tensor encoding Equation 5.13 is adjusted with a closeness strength as follows:

$$\mathcal{X} = \sum_{n=1}^{N} \mathcal{X}_{T_n} = \sum_{T_n=(e_i,r_j,e_k),n=1...N} (\frac{1}{d_{ij}} + \frac{1}{d_{jk}}) \left(I_{e_i} \otimes I_{r_j}^{\top} \right) \otimes I_{e_k}^{\top} \tag{5.15}$$

where d_{ij} and d_{jk} are the distances in text between the pairs (e_i, r_j) and (r_j, e_k) respectively. For instance, the closeness between components of $T_1 = (e_i, r_j, e_k)$=(*garlic*, *kill*, *bacteria*) in the sample sentence "*Garlic can also kill harmful bacteria, fungi, and viruses*" is measured as:

$$\frac{1}{d_{ij}} + \frac{1}{d_{jk}} = \frac{1}{\mid garlic\ kill \mid} + \frac{1}{\mid kill\ bacteria \mid} = \frac{1}{3} + \frac{1}{2} \approx 0.83$$

This value is understood as the closeness strength of components in the triple T_1. The closer the words in the context are, the bigger the values can be. In the case where components of a triple are adjacent, their closeness strength achieves a maximum at 2.

In summary, the semantic space of words can be modelled by using a three-way tensor to capture words in context as triples, and to manage structural information such as word order and syntactic information. This types of representation also offers facilities for further semantic processing tasks.

Computing on Word Meaning

In the earlier sections above, we discussed about the influences of syntagmatic and paradigmatic association on understanding word meaning. This section discusses how

computations on word meanings based on the ideas of syntagmatic and paradigmatic relations can be used. In particular, we suggest how the degree of syntagmatic and paradigmatic association between words can be measured. Based on these measurements, we propose a metric to estimate the semantic distance between words using a three-way tensor.

A Measure of Syntagmatic Association

The degree of syntagmatic association reflects the degree to which two words co-occur more than expected. Using a three-way tensor representation, each word is represented by three different slices in respect to the three tensor modes. In this way, the degree of syntagmatic association between two words is also measured by considering the respective slices in each mode. Given two words e_1, e_2 and their sets of representation slices as $\{\mathcal{X}_{e_1::}, \mathcal{X}_{:e1:}, \mathcal{X}_{::e_1}\}$ and $\{\mathcal{X}_{e_2::}, \mathcal{X}_{:e2:}, \mathcal{X}_{::e_2}\}$, the degree of syntagmatic association between e_1 and e_2 is measured as follows:

$$syn(\mathbf{e_1}, \mathbf{e_2}) = \max\{\frac{\sum_{e_k} \mathcal{X}_{\mathbf{e_1 e_2} e_k} + \sum_{e_i} \mathcal{X}_{e_i, \mathbf{e_1}, \mathbf{e_2}}}{Z_{\mathbf{e_1}}} \ , \ \frac{\sum_{e_k} \mathcal{X}_{\mathbf{e_2 e_1} e_k} + \sum_{e_i} \mathcal{X}_{e_i, \mathbf{e_2}, \mathbf{e_1}}}{Z_{\mathbf{e_2}}}\} \tag{5.16}$$

The value of $syn(\mathbf{e_1}, \mathbf{e_2})$ shows the strength of syntagmatic association between e_1 and e_2. It is calculated based on the sharing fibers of e_1 and e_2. Z_{e_1} and Z_{e_2} are the

normalized factors for each word, and are calculated as:

$$Z_{e_1} = \sum_{(e_j,e_k)} \mathcal{X}_{\mathbf{e_1}e_je_k} + \sum_{(e_i,e_k)} \mathcal{X}_{e_i,\mathbf{e_1},e_k} + \sum_{(e_i,e_j)} \mathcal{X}_{e_i,e_j,\mathbf{e_1}} \tag{5.17}$$

$$Z_{e_2} = \sum_{(e_j,e_k)} \mathcal{X}_{\mathbf{e_2}e_je_k} + \sum_{(e_i,e_k)} \mathcal{X}_{e_i,\mathbf{e_2},e_k} + \sum_{(e_i,e_j)} \mathcal{X}_{e_i,e_j,\mathbf{e_2}} \tag{5.18}$$

Based on the structure of the three-way tensor as well as the method for encoding triples into the tensor, it is possible to infer that the Equation 5.16 can be modified to estimate the nearest neighbours of a word. For instance, given a word e_2, a word e_1 is most likely preceding the word e_2 if it achieves the maximum in the following equation:

$$near(\mathbf{e_1}|\mathbf{e_2}) = argmax\{\frac{\sum_{e_k} \mathcal{X}_{\mathbf{e_1}\mathbf{e_2}e_k} + \sum_{e_i} \mathcal{X}_{e_i,\mathbf{e_1},\mathbf{e_2}}}{Z_{\mathbf{e_2}}}\} \tag{5.19}$$

A Measure of Paradigmatic Association

Two words have a paradigmatic relation if one can be substituted for the other without changing grammaticality and acceptability of the sentence. Moreover, as each of the words is represented by a set of three tensor slices, they are likely to have a strong paradigmatic association if there is a strong correlation between their respective slices. To measure this factor,the Cosine angle between their tensor slices is estimated. Given two words e_1 and e_2, the Cosine angle representing their paradigmatic strength is

measured as:

$$par(\mathbf{e_1}, \mathbf{e_2}) = \frac{\mathcal{X}_{e_1::}.\mathcal{X}_{e_2::} + \mathcal{X}_{:e_1:}.\mathcal{X}_{:e_2:} + \mathcal{X}_{::e_1}.\mathcal{X}_{::e_2}}{\|\mathcal{X}_{e_1::}\mathcal{X}_{:e_1:}\mathcal{X}_{::e_1}\| \ \|\mathcal{X}_{e_2::}\mathcal{X}_{:e_2:}\mathcal{X}_{::e_2}\|} \tag{5.20}$$

$$= \frac{\sum\limits_{(r_j,e_k)} \mathcal{X}_{e_1 r_j e_k} \times \mathcal{X}_{e_2 r_j e_k} + \sum\limits_{(e_i,e_k)} \mathcal{X}_{e_i e_1 e_k} \times \mathcal{X}_{e_i e_2 e_k} + \sum\limits_{(e_i,r_j)} \mathcal{X}_{e_i r_j e_1} \times \mathcal{X}_{e_i r_j e_2}}{\sqrt{\sum\limits_{(r_j,e_k)} \mathcal{X}_{e_1 r_j e_k}{}^2 + \sum\limits_{(e_i,e_k)} \mathcal{X}_{e_i e_1 e_k}{}^2 + \sum\limits_{(e_i,r_j)} \mathcal{X}_{e_i r_j e_1}{}^2} \sqrt{\sum\limits_{(r_j,e_k)} \mathcal{X}_{e_2 r_j e_k}{}^2 + \sum\limits_{(e_i,e_k)} \mathcal{X}_{e_i e_2 e_k}{}^2 + \sum\limits_{(e_i,r_j)} \mathcal{X}_{e_i r_j e_2}{}^2}}$$

A Measure of Semantic Distance

When observing words in context, syntagmatic association reflects category similarity between words, while the paradigmatic association describes words related as synonyms and antonyms. Furthermore, the semantic distance between words is defined as a combination of word relatedness and similarity. Therefore, our metric to measure the semantic distance between words is constructed by using a linear combination of the syntagmatic and paradigmatic strengths. As a result, the semantic distance between two words e_1 and e_2 is measured as follows:

$$dist(e_1, e_2) = \alpha.syn(e_1, e_2) + (1 - \alpha).par(e_1, e_2) \tag{5.21}$$

where α is used as a trade-off parameter between syntagmatic and paradigmatic strengths. This value is identified by scanning in a independent dataset to select the best value before applying it to the testing benchmarks.

In summary, the combination of relation extraction and tensor analysis provides a straightforward way to represent word meanings as well as to measure the semantic distance between words. Different from vector-based representation, our proposed tensor representation maintains both words in context as well as structural information such as syntactical information and word order. These pieces of information contribute to the definition of a metric, which measures the strengths between words in different angles. In particular, we defined a metric for estimating word semantic distance, which is later evaluated on different benchmarks.

5.3.3 Multi-way Latent Feature Analysis

Among many feature generation algorithms developed to analyse word relationships within local contexts, latent feature analysis appears a particular promising method to disclose latent factors for word meaning (Dumais 2004). For example, LSA reveals word-word relationships by disclosing latent factors based on their occurrences within documents (Deerwester et al. 1990), while HAL utilises co-locations between two words in context to figure out latent feature representations of words (Lund & Burgess 1996). These methods have been accomplished by using the techniques of matrix factorization such as Singular Vector Decomposition (Golub & Van Loan 1989). Inspired by the idea, we have adopted tensor decomposition techniques to disclose latent factors from triple representations on a three-way tensor structure.

Parallel Factor Analysis

Parallel factor analysis (PARAFAC) is a multi-linear analogue of the singular value decomposition used in latent semantic analysis (Harshman 1970, Carroll & Chang 1970). It decomposes an original tensor into a model that is multiplied by a matrix along each mode (see Figure 5.5).

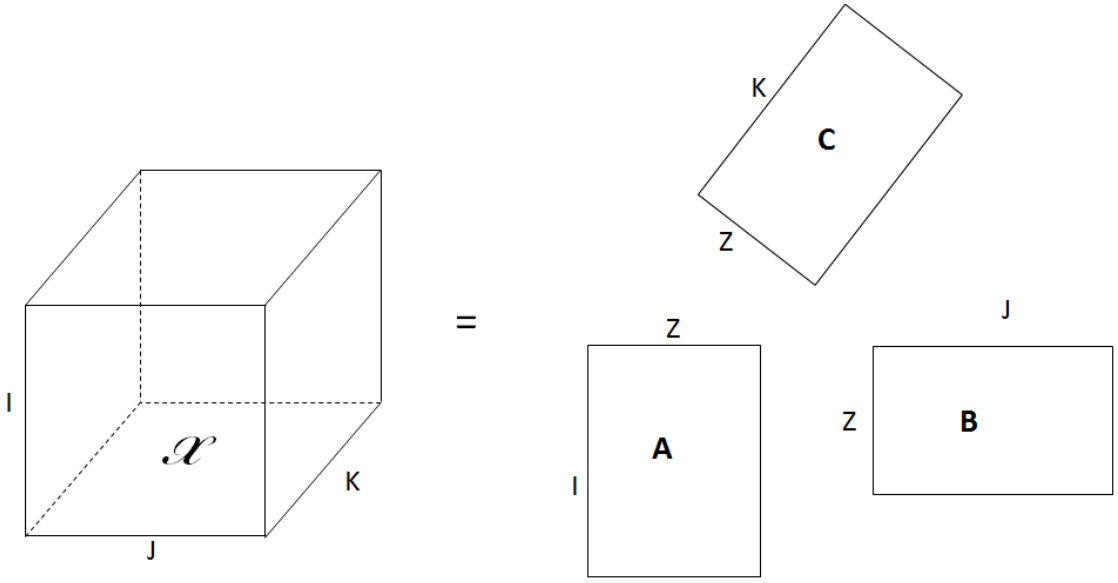


Figure 5.5: Graphical representation of PARAFAC. A tensor $\mathcal{X}$ is approximated into a combination of three loadings matrices: $A_{I\times Z}$, $B_{J\times Z}$, and $C_{K\times Z}$

The key idea of PARAFAC is to minimize the sum of squares between the original tensor and the factorized model of the tensor. This is shown in Equation 5.22 with the case of a three-way tensor $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$.

$$\min_{x_i \in \mathbb{R}^I, y_i \in \mathbb{R}^J, z_i \in \mathbb{R}^K} \| \mathcal{X} - \sum_{i=1}^{Z} x_i \odot y_i \odot z_i \| \tag{5.22}$$

where z is the number of dimensions in the factorized model and $\odot$ is denoted the outer product.

To minimize Equation 5.22, the PARAFAC algorithm finds a tensor (as a sum of Z rank one tensors) that is as similar as possible to the original tensor in the least square sense. The algorithm results in three matrices A ($I \times Z$), B ($J \times Z$), and C ($K \times Z$) that indicate the loading of each mode on the factorized tensor. The model is represented graphically in Figure 5.5. From the meaning representation perspective, each row of the matrix is considered as a Z-dimensional vector of latent features that represents a word in each mode of the tensor.

Representing Word Meaning

Given a triple (e_i,r_j,e_k) encoded into a three-way tensor $\mathcal{X}$, after decomposing $\mathcal{X}$ into three matrices A, B, C, the vector representation of e_i is $A(e_i, :)$ in respect to the first model of the tensor. Similarly, $B(r_j, :)$ and $C(e_k, :)$ are vector representations of r_j and e_k in the second and third modes, respectively. These vectors include Z latent features induced from the tensor structure after the decomposition process. Moreover, given that a word can appear in different positions in a triple, the word is also represented by each vector in each possible mode. Therefore, the aggregate representation of the word is estimated by concatenating its vector representation.

As a result, a vector representation of a word w using decomposed matrices A, B, C of the tensor X is modelled as:

$$vec(w) = A(w,:) \ \& \ B(w,:) \ \&C(w,:) \tag{5.23}$$

where the symbol & represents for a vector concatenation operator and the vector representation of w has $3 \times Z$ dimensions.

Measuring Semantic Distance

After the tensor analysis process, each word in a text corpus is represented by a vector of latent features. To estimate the semantic distance between two words w_1 and w_2, we directly measure the Cosine similarity between the vector representations $vec(w_1)$ and $vec(w_2)$ as follows:

$$\begin{aligned} dist(w_1, w_2) &= \frac{vec(w_1).vec(w_2)}{\|vec(w_1)\|\|vec(w_2)\|} \\ &= \frac{\sum_{i=1}^{n} vec(w_1)_i \times vec(w_2)_i}{\sqrt{\sum_{i=1}^{n} [vec(w_1)_i]^2} + \sqrt{\sum_{i=1}^{n} [vec(w_2)_i]^2}} \end{aligned} \tag{5.24}$$

where the n is the dimension of the vectors, which is equal to $3 \times Z$, and $vec(w_1)_i$ and $vec(w_2)_i$ are the i^{th} component of the respective vectors. This metric is then tested on different benchmarks of semantic distance.

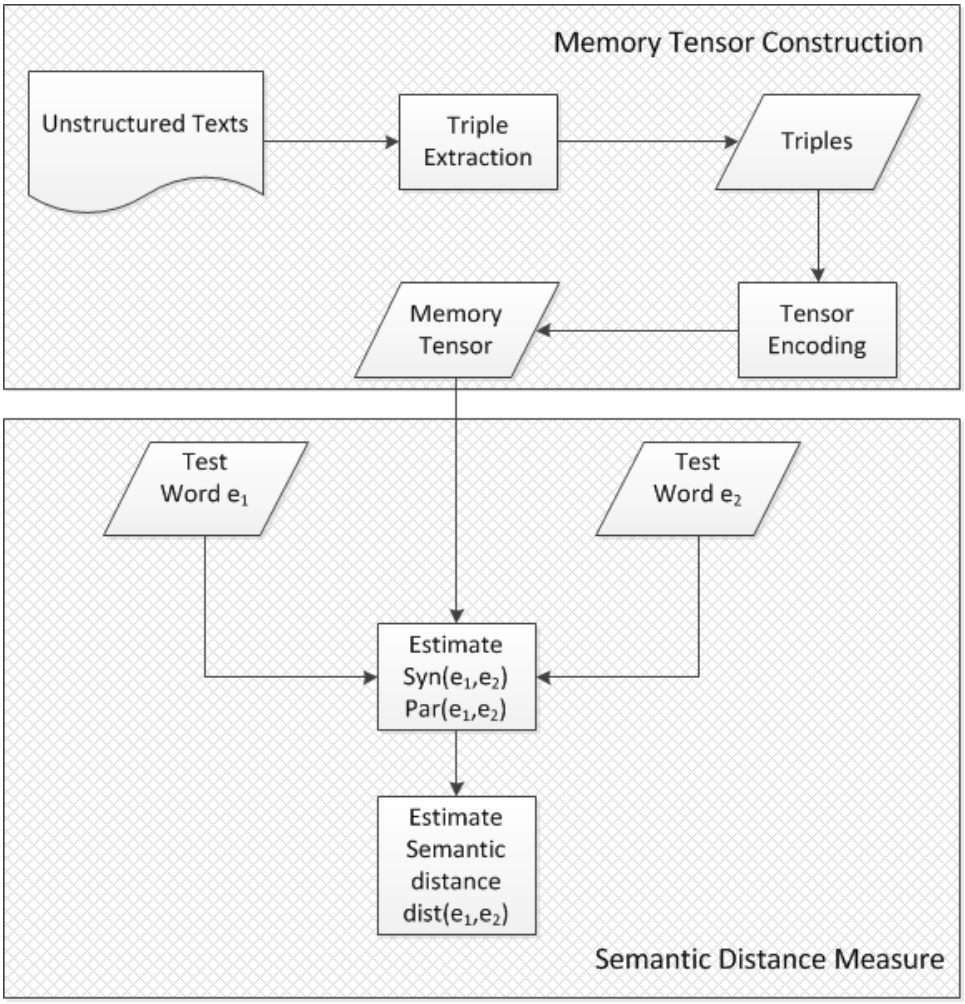


Figure 5.6: Procedures to conduct an experiment for the first Tensor model

5.4 Experiment

Previous sections in this chapter have presented two tensor-based models of representing word meaning. In the first model, structural information from a series of triples was encoded into a three-way tensor, which facilities the semantic processing. In this model, a metric of semantic distance was presented, which takes into account a com-

bination of syntagmatic and paradigmatic measures. Figure 5.6 demonstrates how to construct this model and evaluate on semantic distance measure. In the second model, the meaning representation of words was constructed using latent features induced from a tensor decomposition process. A metric to measure the semantic distance between words was also proposed and tested on different semantic distance benchmarks. Figure 5.7 shows the procedures to conduct this experiment. In this section, we present procedures for conducting an experiment using the two proposed models above as well as evaluating their performance on semantic distance measures.

5.4.1 Text Repository

Our proposed models belong to the category of content-based methods, and are operated on unstructured text data. To be consistent with previous chapters, we used the same text corpus as Wikipedia corpus version October 01, 2012. In particular, we used the first $1,000,000$ pure text articles, which have been parsed from Wikipedia XML dump using Wikiprep[1].

5.4.2 Extracting Relations as Triples

The pure Wikipedia text from each article was fed into a rule-based extractor with confidence (see Section 3.2.1) to return a set of triples. We used the same configuration for the relation extractor (see details in Section 3.5.5), obtaining $36,764,371$ unique

[1]Wikiprep http://sourceforge.net/projects/wikiprep

triples from the entire $1M$ articles used. As the proposed tensor works on the word level, components of each triple were optimized by keeping only the main word in each component; Thus a noun phrase became a single main noun, or a verb phrase was transformed into a main verb. For instance, the original triple ("*Elephant garlic*, "*help to prevent*", "*harmful bacteria*") was optimized to ("*garlic*, "*prevent*", "*bacteria*). After transforming all of the extracted triples, we finally obtained $13,281,760$ unique optimized triples, which were ready for encoding into a three-way tensor.

5.4.3 Encoding Triples into Three-way Tensor

A set of optimized triples was then used to construct a three-way tensor. Using Equation 5.15, each triple ,in turn, was encoded to enrich the tensor. Although each optimized triple does not contain information about its word closeness, when mapping back to its original triples as well as the sentence from where it was extracted, a word closeness value was computed on the fly. Due to the large size of the word indexes (approximately $15,000$) as well as the extreme sparse tensor, we applied the sparse tensor implementation for our experiment (Bader et al. 2012). After all of the optimized triples were encoded, the tensor was ready to measure the semantic distance between words as shown in Equation 5.21.

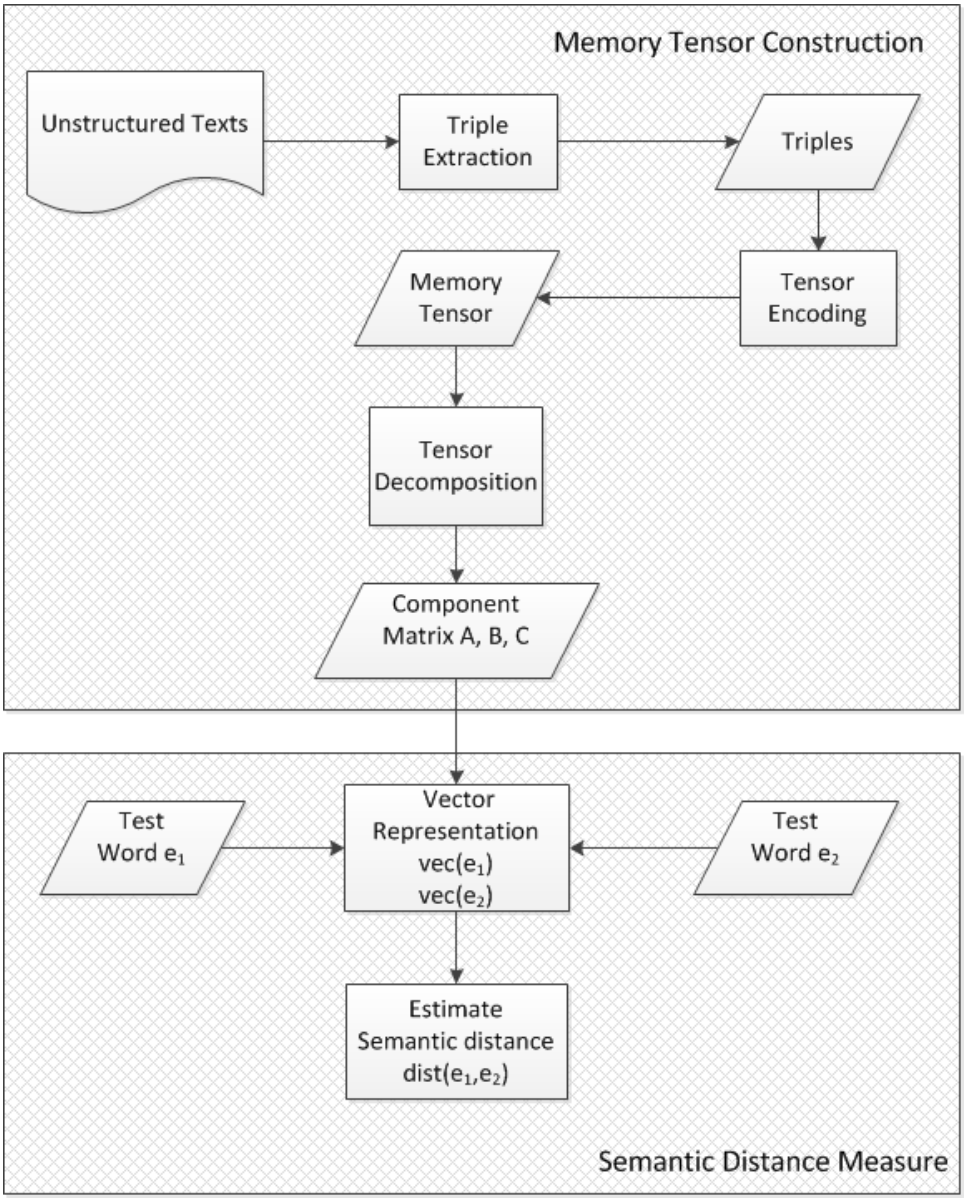


Figure 5.7: Procedures to conduct an experiment for the second Tensor model

5.4.4 Measuring Semantic Distance based on Syntagmatic and Paradigmatic Strengths

Equation 5.21 emphasized that the semantic distance between two words is determined by the combination of their syntagmatic and paradigmatic strengths. Their

influences are determined by the trade-off parameter $\alpha \in [0 \dots 1]$, which needs to be estimated before hand. To have the equal affects of these strengths, we chose $\alpha = 0.5$. However, in our experiment, Table 5.2 clearly shows that the value $\alpha = 0.6$ was able to achieve the best correlation result on the semantic distance measurement tested on the independent test, MTurk. This value was then applied to other testing benchmarks such as WS-353 and RG-65. The table also shows that $\alpha = 0.0$ and $\alpha = 1.0$ are the cases where the semantic distance of words uses only either paradigmatic strengths or syntagmatic strength, respectively.

Table 5.2: Experiment on MTurk for turning the α parameter. The best Spearman's correlation score was obtained with $\alpha = 0.6$

Parameter α	Correlation Results ($\rho \times 100$)
0.0	58.5
0.1	60.1
0.2	62.8
0.3	63.8
0.4	64.6
0.5	64.2
0.6	**65.4**
0.7	65.2
0.8	64.3
0.9	62.1
1.0	59.6

5.4.5 Multi-way Latent Feature Analysis

To extract tensor-based latent features for word meaning representation, we used a tensor decomposition technique (Bader et al. 2012) to find the best fit for the

original tensor. We chose $Z = 300$ as the number of latent features for meaning representation. When the PARAFAC algorithm was minimized as in Equation 5.22, an approximated factorized tensor was returned, which was also the sum of Z rank one tensors. The resulting matrices from the decomposition algorithm are used to form a tensor-based meaning representation of words using Equation 5.23. This model of semantic interpretation was then evaluated using the task of the semantic distance measure (see Equation 5.24) on WS-353 and RG-65 benchmarks.

5.5 Evaluation and Discussion

Two content-based models for semantic interpretation have been proposed and implemented. To evaluate the models, we observed their performances on the task of measuring semantic distance, which was performed on three different benchmarks MTurk (Radinsky et al. 2011), WS-353 (Finkelstein et al. 2001) and RG-65 (Rubenstein & Goodenough 1965) (see Section 2.3.1 for more details about these datasets). While the MTurk dataset was used as a development dataset to find the best α parameter for testing purposes. The other two datasets were used to evaluate the performance of our proposed models.

5.5.1 Overall Results

First of all, Table 5.3 shows the experimental results of each of the proposed model tested on 353 word pairs of WS-353 dataset. The results were also directly compared to the popular baseline algorithm, LSA, on the same benchmark. For the first model

Table 5.3: Correlation results for different tensor-based features used on the semantic distance measure, and tested on the WS-353 dataset using Spearman's rank correlation (ρ). Symbol ‡ indicates value of the α parameter selected from a independent test. Symbol $\star$ indicates the best results over a range of selected parameters. The result from the LSA baseline method is also presented.

Algorithms	$\rho \times 100$
Latent Semantic Analysis(LSA) (Finkelstein et al. 2001)	58.10
Tensor-based Syntagmatic Strength ($\alpha = 0$)	66.27
Tensor-based Paradigmatic Strength ($\alpha = 1.0$)	69.35
Combination Model‡ ($\alpha = 0.6$)	72.53
Combination Model* ($\alpha = 0.7$)	**74.28**
Tensor-based Latent Features	67.18

that based on syntagmatic and paradigmatic strengths, the correlation results with the development parameter $\alpha = 0.6$ shows a significant improvement compared to the baseline benchmarks (72.53% vs. 58.10%). With the second model that focusing on latent features in the tensor structure, the proposed feature generation also returned outstanding results on the semantic distance task compared to the baseline one (67.18% vs. 58.10%). These numeric assertions emphasize the effects of modelling unstructured text using the tensor structure.

Secondly, while the design of WS-353 dataset tends to emphasize the semantic relatedness between words, RG-65 focuses on semantic similarity. Table 5.4 illustrates the correlation results when applying the proposed models to measure the semantic distances between 65 word pairs from the RG-65 dataset. With the development parameter of $\alpha = 0.6$, the combination model achieved promising results (79.87%) on this dataset, which was also the case when compared to the baseline benchmark

Table 5.4: Correlation results for different tensor-based features used on the semantic distance measure, and tested on the RG-65 dataset using Spearman's rank correlation (ρ). Symbol ‡ indicates value of the α parameter selected from a independent test. Symbol $\star$ indicates the best results over a range of selected parameters. The result from the LSA baseline method is also presented.

Algorithms	$\rho \times 100$
Latent Semantic Analysis(LSA) (Finkelstein et al. 2001)	60.09
Tensor-based Syntagmatic Strength ($\alpha = 0$)	79.39
Tensor-based Paradigmatic Strength ($\alpha = 1.0$)	77.69
Combination Model‡ ($\alpha = 0.6$)	79.87
Combination Model* ($\alpha = 0.3$)	**83.27**
Tensor-based Latent Features	78.62

(60.09%). Moreover, although showing slightly lower results compared to the combination model, the tensor-based feature generation model also performed better in relation to the LSA baseline method. This again confirms that whether tested on relatedness-bias or similarity-bias datasets, our proposed model of semantic interpretation has distinct advantages over the baseline model.

5.5.2 Syntagmatic Strength vs Paradigmatic Strength

Notably, when using the best parameter selected from the development dataset, none of our models achieved the best results on the two testing benchmarks. Instead, the first model achieves the best result at $\alpha = 0.7$ in the WS-353 datasets and $\alpha = 0.3$ in the RG-65. One explanation for this could derive from the nature of the dataset created as well as the facts behind the value of α. As it is designed, when the value of $\alpha > 0.5$, the strength of syntagmatic relations had more influence than that of

paradigmatic relation on the semantic distance measurement (see Equation 5.21). This also indicate that the model tends to produce an appropriate semantic distance on the related pairs of WS-353 dataset. By contrast, when $\alpha < 0.5$, the semantic distance metric was more influenced from paradigmatic relations, and tends to produce an appropriate distance between similar pairs on RG-65. The correlation results from Tables 5.3 and 5.4 also emphasize this trends as the syntagmatic strength ($\alpha = 0$) is dominant over paradigmatic strength ($\alpha = 1$) on WS-353 and the reverse situation is found in RG-65.

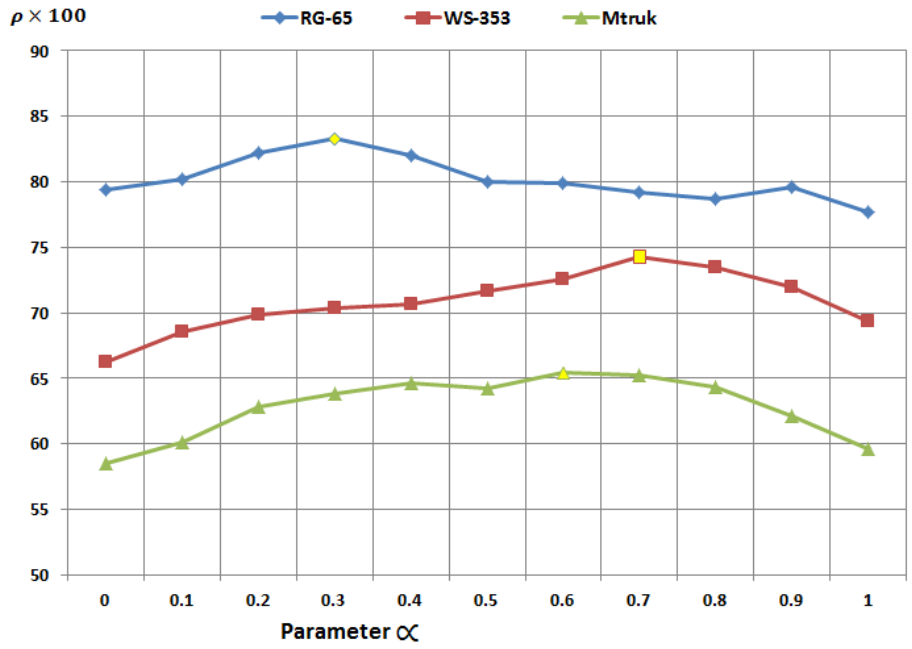


Figure 5.8: Correlation results (%) on each datasets have been presented when changing the trade-off parameter α

Figure 5.8 shows another view when changing α values in each dataset. The best

correlation with human judgements was achieved with different influences of syntagmatic strengths and paradigmatic strengths. There was also a strong correlation between the results from WS-353 and MTurk as both datasets contain dominant related pairs rather than similar pairs.

5.5.3 Further remarks

In each of the proposed methods, word order information is used for encoding text surface into tensor structure. Moreover, it is also used (1) explicitly on the semantic distance measurement based on syntagmatic and paradigmatic strengths, and (2) implicitly to induce latent features using tensor decomposition techniques. The performance of these models was significantly improved when compared with the baseline methods.

Syntagmatic and paradigmatic strengths performed differently in measuring the distance between words. However, when combined into a single model, there was a significant improvement compared to when they were used separately.

Syntactical information has also played an important role. With the use of syntactic analysis to extract information on triples, a significant amount of qualified relations were maintained while eliminating a large amount of non-essential information. This helps significantly in reducing the size of the word dictionary as well as the resources used to construct the three-way tensor.

To demonstrate the effectiveness of the proposed method, we only used triple

structure to store lexical information. However, it is possible to use other types of word-word relationships such as syntactic dependencies, or semantic relations between words to figure out meaning representation as well as to measure semantic distance between words.

There are challenges when dealing with large amount of text data. In our experiment, we only used a part of the Wikipedia repository to evaluate the proposed models. However, when using the full data repository or processing large amount of text information on the Web, a special care should be taken to reduce the explosion of dimensions, especially when performing tensor analysis to disclose latent features. This is why a tensor-based model of generating latent features is more appropriate in processing small text domains.

5.6 Conclusion

In this chapter, we have presented content-based methods to represent word meanings and to measure the semantic distance between words. The first method utilizes structural information from extracted triples, which are encoded into a three-way tensor structure. The semantic distance between words is estimated based on the idea of syntagmatic and paradigmatic associations. In the second method, we utilize tensor decomposition techniques to disclose latent features from a three-way tensor. These methods are evaluated using semantic distance measure over different standard benchmarks. The experimental results show the strong correlation between our proposed

metrics and human judgements. They also indicate a significant improvement when compared to results of popular baseline methods testing on the same benchmarks.

Chapter 6
CONCLUSION

This chapter concludes the work done in this thesis as well as presenting a discussion of future directions. We begin with a summary of the proposed approaches, and then the effects of each individual approach are analysed based on the results from measuring semantic distance. We also address the research questions raised in the earlier sections, some limitations of this research, and conclude with a discussion of future directions.

6.1 Summaries of the Proposed Approaches

Early in Chapter 1, we argued for the needs of content-based approaches for semantic interpretation. We looked at fundamental problems and proposed three different aspects of feature generation for word meaning representation that would benefit for investigation. For each aspect, an equivalent approach has been proposed.

For the first approach, the main idea was to consider lexical syntactic relationships between words in contexts in order to understand the meaning representation of words. We introduced a relation extraction technique to extract triples representing relationships between words under the template "(*entity, relation, entity*)". We also introduced a confidence model, which assigned a confidence value for each triple

extraction. This model was used as a filtering technique to retain qualified triples for word meaning representation. A vector representation for word meaning was created in two ways: (1) using triple components directly as the features where the weighting schema was measured using point-wise mutual information on the word frequencies within the triples; and (2) hidden concepts induced as feature representation of the vector. To evaluate the effects of these kinds of the features, we implemented a task of measuring semantic distance between words using WS-353 and RG-65 as testing benchmarks.

Next, in the second approach, we looked at the contribution of topical information on meaning representation and undertook several measurements to find the best combination between topical model and context word features. The ideas behind this approach was to show that meanings in contexts of a word is influenced not only by its distributions in local contexts but also in terms of the global contexts such as the themes, genres, or topics where the word appears. Therefore, two major steps were investigated to yield word meaning representation:

(1) From the contexts where a word appeared, its global topics needed to be induced. We implemented a topical model based on LDA to construct latent topics from unstructured texts.

(2) To find a suitable combination of multiple aspect features such as word features and latent topic features. We implemented several feature combinations and found suitable ways to integrate these kinds of features to form word meaning representation.

We used a VSM to hold the final representation of words, which were tested by the task of measuring semantic distance.

For the third approach, we aimed to construct representation of a word that takes into account not only words in context, but also structural information such as word order and syntactic aspects. To do so, we used a triple structure which enable the maintenance of relations between words in contexts as well as the management of syntactic relations between triple components. Information from triples was then encoded into a three-way tensor, which was able to hold context content as well as structural information. We also embedded word closeness aspects into triple weightings, which emphasized the strength of each triple depending on their positions in the text. By using a three-way tensor for encoding contextual information, a word in contexts was represented by each tensor slice for each tensor mode. To exploit the structural information from the encoded tensor, we borrowed the idea of syntagmatic and paradigmatic relations, which heavily depend on word order and syntactic aspects. We evaluated the effects of this meaning representation by measuring the semantic distance between words. Moreover, we further processed information encoded in the tensor to disclose latent features, which were then used to form the representation of words, and further tested by measuring the semantic distance.

Table 6.1: A summary of the best results of our proposed features tested on the WS-353 dataset for the task of semantic distance measurement.

Features used in vector representation of words	$\rho \times 100$
Representation using a vector of relational features	72.25
Representation using a vector of relational hidden features	71.16
Latent Topic Features	67.01
Concatenating combination between relational features and latent topic features	73.6
Mixture combination between relational features and latent topic features	71.92
Tensor-based combination features	74.28
Tensor-based latent features	67.18

6.2 Overall Results from Measuring Semantic Distance

To evaluate the effectiveness of the proposed approaches, the task of measuring semantic distance was used. We selected three standard datasets to measure the performance of the proposed approaches. In each dataset, human judgement scores are considered as the gold standards, and were used to compare against the results generated from each of the approaches. The correlation results from these comparisons were recorded to determine the effectiveness of our proposed approaches.

Information from Table 6.1 briefly summaries the overall results for the proposed features tested on WS-353. Meanwhile, the correlation results on RG-65 is also shown in Table 6.2. Most of the proposed features shows significant improvement compared to the baseline benchmarks as well as showing comparable results with most related approaches on the same testing benchmarks (refer to each chapter for more detail).

This gives an overall view of the performance of the approaches on different feature aspects.

Table 6.2: A summary of the best results of our proposed features tested on the RG-65 dataset for the task of semantic distance measurement.

Features used in vector representation of words	$\rho \times 100$
Representation using a vector of relational hidden features	85.31
Representation using a vector of relational features	85.75
Latent topic features	63.93
Concatenating combination between relational features and latent topic features	84.62
Mixture combination between relational features and latent topic features	82.17
Tensor-based combination features	83.27
Tensor-based latent features	78.62

6.3 Addressing Research Questions

Several research questions were raised in Chapter 1. In this section, we aim to address these questions based on the findings and results from our experiments.

6.3.1 What strategies can be used to utilise relations between words for semantic interpretation?

As the nature of human language, words appearing together in contexts may have certain interactions with surrounding words. Such relations can be based on syntactic, lexical as well as semantic aspects. In our approach (see Chapter 3), we assume that relations between words in contexts can be managed by using a triple structure, and

therefore the semantic representation of a word is constructed by considering all of the triples containing that word. In this approach, the representation of a word is a high-dimensional vector, where each relational word becomes a feature representation. Given this type of organization, the representation of a word works effectively via the task of measuring semantic distance. However, there is an existing drawback from this strategy. That is the high-dimensional vectors from the word representation, which can affect the performance from adopted applications. One possible solution, that has been conducted to address the issue, was to perform a dimension reduction technique using SVD. This involved a slight trade-off in terms of reliability but was still acceptable within our experiment.

6.3.2 What are the effects of topical analysis for word meaning representation?

In Chapter 4 of the thesis, we analysed the influences of topical information on word meaning representation. Topical information itself was induced directly from text surface. The meanings of a word in contexts were represented by a vector of latent topics. This modelling was tested on the task of measuring semantic distance. Its performance was acceptable as it was still better than traditional baseline benchmarks. However, when compared to a method that uses non-topical information, the performance of topical information was still under expectation. This was why we proposed a method to combine topical features and features from words in context. Two typical feature combinations were performed: (1) linear combination of topical

features and word context features, and (2) mixture combination of two types of features based on the seeding information for each topic. While the first combination produced outstanding results on semantic distance tests, the second showed significant improvement compared to the use of topical features alone. However, the trade-offs were resulted from the first combination as the high dimensional vectors, and from the second combination as a slight decrease of correlation results Overall, the performance of topical information for meaning representation was promising through out our experiment.

6.3.3 What are the strategies to use words in contexts and structural information on the task of semantic interpretation?

In Chapter 5, we proposed a method to encode words in contexts and structural information into a three-way tensor, which could be used to construct semantic representation of words as well as to measure semantic distance. Words that appeared in contexts were captured using the structure of triples where word orders and syntactical relations between words are maintained. The use of tensor structure offers a straightforward way to access relations between words in contexts as well as their structural information. For instance, in the task of semantic distance, the ideas of syntagmatic and paradigmatic strengths have been used to access information from the tensor as well as to measure the semantic distance between words. The promising results from our experiments confirmed the positive contribution of tensor analysis

as well as the use of structure information on semantic interpretation and semantic distance. However, there are certain limitations when using a multi-way tensor. Large amounts of memory are needed to store information on the tensor. Also, the task of decomposing a tensor into a sum of multiple rank-one tensors is also expensive when using a large tensor. It is thus best to reduce the size of the tensor representation or use the approach with a reasonably sized text domain.

6.3.4 What strategies can be used to improve the performance of content-based methods on the task of measuring semantic distance?

Methodologies from Chapters 3, 4, and 5 have revealed promising ways to improve the correlation results compared to popular methods such as LSA. They reduce the gaps in performance between content-based methods and knowledge-based methods on the task of semantic distance measurement. However, with the content-based methods, it is essential to process large amounts of text data to maintain the performance on different text domains. This is also an enormous challenge that must be dealt with to improve robustness in content-based methods.

6.4 Future Investigations

Our proposed approaches would benefit from further investigation that could reduce their limitations as well as expand their potential.

Our proposed method used the triple structure to hold the relationship between words in context. The method worked extensively well compared to traditional ones

which use other mechanism to retrieve relations of word pairs. However, we only use lexical relations based on basic syntactic structure such as *(subject, verb, object), (noun, preposition, noun)*. It would be worth exploring how to extend the model to different kinds of relations such as syntactic dependencies or semantic relations. Moreover, to demonstrate the proposed methods are workable, we were limited to word with pure words in text rather than to real semantic entities such as a person's name, or a location. Models that consider these mentions could extend the coverage and the uses of text data, and could bring new phases for semantic interpretation.

Secondly, our approaches mainly focused on discovering the semantics of individual words in contexts rather than the meanings of a sequence of texts. Although they have the potential to extend the representation of a word to a sequence of words, there is also a need to extend the models to induce semantic representation of texts from scratch.

Finally, some of the proposed models were relied on heavily processing tasks such as tensor decomposition, and topic induction. It would be necessary to extend the model to have capacity to work on text domains of different sizes as well as languages.

6.5 Future Applications

When delivering a way to model the semantic representation of words as well as measuring semantic distances, plenty of scopes for NLP-based applications can benefit. Some application directions are worthy of notes as the followings.

Document classification is a traditional task of assigning a document to one or more classes depending on its contents. One of the key techniques lead the task successfully is to determine the topical similarity between a pair of documents. Traditional methods for doing this are to consider each document as a "bag of words". The distance between these two would be determined by the degree of overlapping between two "bags of words" using a certain weighting mechanism. In other words, the task of measuring the distance between two documents is broken down into the collection of measuring distance between words individually. This becomes problematic when there is limited overlap between two "bags of words", and even worse when the "bags of words" from two documents have nothing in common. In this situation, the traditional method for measuring the distance between two documents (two pieces of text) has limited success. In this case, either of our proposed methods on semantic distance measurement offer an alternative way for measuring the distance. Given two pieces of text, the semantic distance measurement could produce the distance between their meanings regardless of the overlapping of their word surfaces.

Text Search is an application which retrieves results based on searching conditions compacted in a query. Its search engine normally receives input queries as a set of keywords and then needs to return the search results as relevant documents related to the searching conditions. There are variety search engines ranging from the simple to the complex one, but the core fundamental procedures are quite similar: (1) Matching the input query to relevant documents and (2) Ranking search results. Traditional

search engines tackle the first step by using techniques of keyword matching, where the relevant documents are identified by the co-occurrence with keywords in the input query. However, with the use of our proposed semantic distance measure between the input query and documents, the search engines have the potential to improve matching and ranking policies through a better understanding of the searcher intent and the contextual meanings of keywords in the input query. This is also a target of a semantic search engine.

Sentiment analysis (also known as opinion mining) is another way to use text analysis to determine the attitudes of a speaker/writer with respect to certain topics or the overall contextual polarity of a document. For instance, when reading user reviews of a certain product, the user's attitude can be classified into "negative", "neutral", or "positive" categories. In each category, there are various ways to use natural language to describe a reviewer's attitude. For instance, from a restaurant review, customers are likely use different means to give positive feedback such as "*the food was excellent*", "*the price was not too bad*", "*the service was acceptable*". Given the complexity of language used to describe attitudes, traditional word matching is limited in determining the polarity of a particular piece of text. The uses of our proposed semantic distance measurement would better disclose the user attitudes in the review as well as to compare and contrast with other reviews. This could also determine how close a piece of text is to a particular attitude category.

6.6 Conclusion

In this thesis, we have investigated multiple content-based approaches to address the problems of semantic representation and semantic distance. The approaches utilises unstructured text content to the model meaning representation of words and to measure the semantic distance between them. Each of the approaches focuses on a particular aspect of the problem. While the first approach investigated the potential uses of word relations in meaning representation, the second approach focuses on topical information as well as its combination with non-topical features. The third approach concentrated on combining words in contexts with related structural information such as word order and syntactical aspects. To evaluate the effects of each approach, we conducted the task of measuring semantic distance between words using popular testing benchmarks. Each of the proposed approaches demonstrates positive results which have real potential for future applications.

BIBLIOGRAPHY

Acar, E. & Yener, B. (2009), 'Unsupervised multiway data analysis: A literature survey', *Knowledge and Data Engineering, IEEE Transactions on* **21**(1), 6–20.

Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Paşca, M. & Soroa, A. (2009), A study on similarity and relatedness using distributional and wordnet-based approaches, *in* 'Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics', Association for Computational Linguistics, pp. 19–27.

Bader, B. W., Kolda, T. G. et al. (2012), 'Matlab tensor toolbox version 2.5', Available online.
URL: *http://www.sandia.gov/ tgkolda/TensorToolbox/*

Baroni, M. & Lenci, A. (2010), 'Distributional memory: A general framework for corpus-based semantics', *Computational Linguistics* **36**(4), 673–721.

Bengio, Y., Schwenk, H., Senécal, J.-S., Morin, F. & Gauvain, J.-L. (2006), Neural probabilistic language models, *in* 'Innovations in Machine Learning', Springer, pp. 137–186.

Blei, D. M., Ng, A. Y. & Jordan, M. I. (2003), 'Latent dirichlet allocation', *the Journal of machine Learning research* **3**, 993–1022.

Budanitsky, A. & Hirst, G. (2006), 'Evaluating wordnet-based measures of lexical semantic relatedness', *Computational Linguistics* **32**(1), 13–47.

Carroll, J. D. & Chang, J.-J. (1970), 'Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition', *Psychometrika* **35**(3), 283–319.

Collobert, R. & Weston, J. (2008), A unified architecture for natural language processing: Deep neural networks with multitask learning, *in* 'Proceedings of the 25th international conference on Machine learning', ACM, pp. 160–167.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. & Kuksa, P. (2011), 'Natural language processing (almost) from scratch', *The Journal of Machine Learning Research* **12**, 2493–2537.

Dagan, I., Marcus, S. & Markovitch, S. (1993), Contextual word similarity and estimation from sparse data, *in* 'Proceedings of Association for Computational Linguistics', Association for Computational Linguistics, pp. 164–171.

De Lathauwer, L., De Moor, B., Vandewalle, J. & by Higher-Order, B. S. S. (1994), Singular value decomposition, *in* 'Proc. EUSIPCO-94, Edinburgh, Scotland, UK', Vol. 1, pp. 175–178.

De Saussure, F. (1916/1996), 'Cours de linguistique gnrale', *Paris: Payot* .

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W. & Harshman, R. A. (1990), 'Indexing by latent semantic analysis', *JASIS* **41**(6), 391–407.

Dinu, G. & Lapata, M. (2010), Measuring distributional similarity in context, *in* 'Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, pp. 1162–1172.

Dumais, S. T. (2004), 'Latent semantic analysis', *Annual Review of Information Science and Technology* **38**(1), 188–230.

Etzioni, O., Banko, M., Soderland, S. & Weld, D. S. (2008), 'Open information extraction from the web', *Communications of the ACM* **51**(12), 68–74.

Fader, A., Soderland, S. & Etzioni, O. (2011), Identifying relations for open information extraction, *in* 'Proceedings of the Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, pp. 1535–1545.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G. & Ruppin, E. (2001), Placing search in context: The concept revisited, *in* 'Proceedings of the 10th international conference on World Wide Web', ACM, pp. 406–414.

Gabrilovich, E. & Markovitch, S. (2007), Computing semantic relatedness using wikipedia-based explicit semantic analysis., *in* 'IJCAI', Vol. 7, pp. 1606–1611.

Gabrilovich, E. & Markovitch, S. (2009), 'Wikipedia-based semantic interpretation for natural language processing', *Journal of Artificial Intelligence Research* **34**(2), 443.

Golub, G. H. & Van Loan, C. (1989), 'Matrix computations'.

Harris, Z. S. (1954), 'Distributional structure.', *Word* **10**, 146–162.

Harshman, R. A. (1970), 'Foundations of the parafac procedure: models and conditions for an" explanatory" multimodal factor analysis'.

Hassan, S. & Mihalcea, R. (2011), Semantic relatedness using salient semantic analysis., *in* 'AAAI'.

Hirst, G. & St-Onge, D. (1998), 'Lexical chains as representations of context for the detection and correction of malapropisms', *WordNet: An electronic lexical database* **305**, 305–332.

Huang, E. H., Socher, R., Manning, C. D. & Ng, A. Y. (2012), Improving word representations via global context and multiple word prototypes, *in* 'Proceedings of Association for Computational Linguistics', Association for Computational Linguistics, pp. 873–882.

Huynh, D., Tran, D. & Ma, W. (2014), Combination features for semantic similarity measure, *in* 'Proceedings of the International MultiConference of Engineers and Computer Scientists', Vol. 1.

Jiang, J. J. & Conrath, D. W. (1997), 'Semantic similarity based on corpus statistics and lexical taxonomy', *f International Conference Research on Computational Linguistics* .

Jones, M. N. & Mewhort, D. J. (2007), 'Representing word meaning and order information in a composite holographic lexicon.', *Psychological review* **114**(1), 1.

Kolda, T. G. & Bader, B. W. (2009), 'Tensor decompositions and applications', *SIAM review* **51**(3), 455–500.

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

Landauer, T. K., Foltz, P. W. & Laham, D. (1998), 'An introduction to latent semantic analysis', *Discourse processes* **25**(2-3), 259–284.

Leacock, C. & Chodorow, M. (1998), 'Combining local context and wordnet similarity for word sense identification', *WordNet: An electronic lexical database* **49**(2), 265–283.

Lin, D. (1998*a*), Automatic retrieval and clustering of similar words, *in* 'Proceedings of the 17th international conference on Computational linguistics-Volume 2', Association for Computational Linguistics, pp. 768–774.

Lin, D. (1998*b*), An information-theoretic definition of similarity., *in* 'ICML', Vol. 98, pp. 296–304.

Lund, K. & Burgess, C. (1996), 'Producing high-dimensional semantic spaces from lexical co-occurrence', *Behavior Research Methods, Instruments, & Computers* **28**(2), 203–208.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D. & Miller, K. J. (1990), 'Introduction to wordnet: An on-line lexical database', *International journal of lexicography* **3**(4), 235–244.

Miller, G. A. & Charles, W. G. (1991), 'Contextual correlates of semantic similarity', *Language and cognitive processes* **6**(1), 1–28.

Mnih, A. & Hinton, G. (2007), Three new graphical models for statistical language modelling, *in* 'Proceedings of the 24th international conference on Machine learning', ACM, pp. 641–648.

Perfetti, C. A. (1998), 'The limits of co-occurrence: Tools and theories in language research'.

Phan, X.-H., Nguyen, L.-M. & Horiguchi, S. (2008), Learning to classify short and sparse text & web with hidden topics from large-scale data collections, *in* 'Proceedings of the 17th international conference on World Wide Web', ACM, pp. 91–100.

Radinsky, K., Agichtein, E., Gabrilovich, E. & Markovitch, S. (2011), A word at a time: computing word relatedness using temporal semantic analysis, *in* 'Proceedings of the 20th international conference on World wide web', ACM, pp. 337–346.

Rapp, R. (2002), The computation of word associations: comparing syntagmatic and paradigmatic approaches, *in* 'Proceedings of the 19th international conference on Computational linguistics-Volume 1', Association for Computational Linguistics, pp. 1–7.

Reisinger, J. & Mooney, R. J. (2010), Multi-prototype vector-space models of word meaning, *in* 'Human Language Technologies: The 2010 Annual Conference

of the North American Chapter of the Association for Computational Linguistics', Association for Computational Linguistics, pp. 109–117.

Resnik, P. (1995), 'Using information content to evaluate semantic similarity in a taxonomy', *Proceedings of IJCAI* .

Resnik, P. (1999), 'Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language', *Journal Of Artificial Intelligence Research* **11**, 95–130.

Rubenstein, H. & Goodenough, J. B. (1965), 'Contextual correlates of synonymy', *Communications of the ACM* **8**(10), 627–633.

Sahlgren, M. (2006), The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces, PhD thesis, Stockholm.

Sahlgren, M., Holst, A. & Kanerva, P. (2008), Permutations as a means to encode order in word space, *in* 'The 30th Annual Meeting of the Cognitive Science Society'.

Salton, G., Wong, A. & Yang, C.-S. (1975), 'A vector space model for automatic indexing', *Communications of the ACM* **18**(11), 613–620.

Schutze, H. (1993), Word space, *in* 'Advances in Neural Information Processing Systems 5', Citeseer.

Schutze, H. & Pedersen, J. (1993), A vector model for syntagmatic and paradigmatic relatedness, *in* 'Proceedings of the 9th Annual Conference of the UW Centre for the New OED and Text Research', Citeseer, pp. 104–113.

Schutze, H. & Pedersen, J. O. (1995), Information retrieval based on word senses, *in* 'In Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval', Citeseer.

Strube, M. & Ponzetto, S. P. (2006), Wikirelate! computing semantic relatedness using wikipedia, *in* 'AAAI', Vol. 6, pp. 1419–1424.

Symonds, M., Bruza, P. D., Sitbon, L. & Turner, I. (2012), A tensor encoding model for semantic processing, *in* 'Proceedings of the 21st ACM international conference on Information and knowledge management', ACM, pp. 2267–2270.

Turney, P. (2007), 'Empirical evaluation of four tensor decomposition algorithms'.

Turney, P. D. (2001), Mining the web for synonyms: Pmi-ir versus lsa on toefl, *in* 'Proceedings of the 12th European Conference on Machine Learning', pp. 491–502.

Turney, P. D. (2006), 'Similarity of semantic relations', *Computational Linguistics* **32**(3), 379–416.

Turney, P. D., Pantel, P. et al. (2010), 'From frequency to meaning: Vector space models of semantics', *Journal of artificial intelligence research* **37**(1), 141–188.

Van de Cruys, T., Poibeau, T., Korhonen, A. et al. (2013), A tensor-based factorization model of semantic compositionality, *in* 'Proceedings of the Conference of the North American Chapter of the Association of Computational Linguistics (HTL-NAACL)', pp. 1142–1151.

Van Rijsbergen, C. J., Robertson, S. E. & Porter, M. F. (1980), *New models in probabilistic information retrieval*, Computer Laboratory, University of Cambridge.

Wikipedia (2004), 'Wikipedia, the free encyclopedia'.
URL: *http://wikipedia.org*

Wu, Z. & Palmer, M. (1994), Verbs semantics and lexical selection, *in* 'Proceedings of the 32nd annual meeting on Association for Computational Linguistics', Association for Computational Linguistics, pp. 133–138.

Yeh, E., Ramage, D., Manning, C. D., Agirre, E. & Soroa, A. (2009), Wikiwalk: random walks on wikipedia for semantic relatedness, *in* 'Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing', Association for Computational Linguistics, pp. 41–49.

Zesch, T., Muller, C. & Gurevych, I. (2008), Using wiktionary for computing semantic relatedness., *in* 'AAAI', Vol. 8, pp. 861–866.