UNIVERSITY
OF TECHNOLOGY
SYDNEY

Reliable Sentiment Analysis in Social Media

by Qian Li

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of Qiang Wu, Jian Zhang

University of Technology Sydney
Faculty of Engineering and Information Technology

July, 2020

CERTIFICATE OF ORIGINAL AUTHORSHIP

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

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

This document has not been submitted for qualifications at any other academic institution. This research is supported by the Australian Government Research Training Program.

Production Note:

SIGNATURE: Signature removed prior to publication.

Qian Li

DATE: 29 July, 2020

Abstract

Sentiment analysis in social media is critical yet challenging because the source materials (i.e., reviews posted in social media) are with high complexity, low quality, and uncertain credibility. For example, words and sentences in a textual review may couple with each other, and they may have heterogeneous meanings under different contexts or in different language locales. These couplings and heterogeneities essentially determine the sentiment polarity of the review but are too complex to be captured and modeled. Also, social reviews contain a large number of informal words and typos (a.k.a., noise) but a rare number of vocabularies (a.k.a., sparsity). As a result, most of the existing natural language processing (NLP) methods may fail to represent social reviews effectively. Furthermore, a large proportion of social reviews are posted by fraudsters. These fraud reviews manipulate social opinion, and thus, they disturb sentiment analysis.

This research focuses on reliable sentiment analysis in social media. It systematically investigates the sentiment analysis techniques to tackle three major challenges in social media: high data complexity, low data quality, and uncertain credibility. Specifically, this research focuses on two research problems: general sentiment analysis in social media and fraudulent sentiment analysis in social media. The general sentiment analysis targets on tackling high data complexity and low-quality of social articles that are credible. The fraudulent sentiment analysis handles the uncertain credibility issue, which is common and profoundly affects the precise sentiment analysis in social media. Based on these investigations, this research proposes a serial of methods to achieve reliable sentiment analysis: It studies the polarity-shift characteristics and non-IID characteristics in general paragraphs to capture the sentiment more accurately. It further models multi-granularity noise and sparsity in short text, which is the most common data in social media, for robust short text sentiment analysis. Finally, it tackles the uncertain credibility problem in social media by studying fraudulent sentiment analysis in both supervised and unsupervised scenarios.

This research evaluates the performance and properties of the proposed reliable sentiment analysis methods by extensive experiments on large real-world data sets. It demonstrates that the proposed methods are superior and reliable in social media sentiment analysis.

DEDICATION

To my son Jewell Zhu…

Acknowledgments

First of all, I want to thank all the contributions my supervisor Prof. Qiang Wu make during my Ph.D experience. I appreciate it very much for his time, ideas, discussions, and paper polishing to make my Ph.D. studies impressive and stimulating. I am deeply influenced by his professionalism, profound ideas and hard work. Also, his optimism attitude and enthusiasm were contagious and inspiring for me, even during the difficult times in the pursuit of my doctoral degree.

I am also greatly indebted to Prof. Jian Zhang, my co-supervisor, whose constant suggestions, incisive comments and encouragements helped me a lot during my Ph.D. studies.

My sincere thanks also go to all of my colleagues in the Global Big Data Technologies Centre, for their enthusiastic help: Dr.Peng Zhang, Dr. Zongjian Zhang, Xunxiang Yao, Dr. Xiaoshui Huang, Yan Huang, Lu Zhang, Yongshun Gong, Xiaofei Liu, Anan Du, Lingxiang Yao. I appreciate our intensive discussions, in-depth collaboration, and all the fun we have had in the last four years.

I thank the colleagues in the student union. They broadened my horizon and shared unforgettable times with me in my study abroad. In particular, I am grateful to Mr. Wenqi Niu, Mrs. Jing Zhao, Mr. Yongfeng Hou, Dr. Haodong Chang, Dr. Yu Suo, Dr. Yu Chen, Dr. Yan Wang, Dr. Kunpeng Zhao and Zhixun Zhao for their precious advice and encouragement in my Ph.D. research.

I gratefully acknowledge the funding from China Scholarship Council. Without its funding, my Ph.D work would not be possible.

Lastly, I would like to thank my family for all their selfless love and support. Without their encouragement, it would not be possible for me to conduct this research. In particular, my grateful thanks goes to my husband Dr. Chengzhang Zhu whose spiritual and academic support are so appreciated. Thank you.

Qian Li
University of Technology Sydney
July 2020

List of Publications

Related to the Thesis :

Published Papers :

1. **Li, Q.**, Wu, Q., Zhu, C., and Zhang, J., 2019, September. Bi-level masked multi-scale CNN-RNN networks for short text representation *In International Conference on Document Analysis and Recognition*. IEEE.
2. **Li, Q.**, Wu, Q., Zhu, C., Zhang, J. and Zhao, W., 2019, July. An inferable representation learning for fraud review detection with cold-start problem. *In 2019 International Joint Conference on Neural Networks*. IEEE.
3. **Li, Q.**, Wu, Q., Zhu, C., Zhang, J. and Zhao, W., 2019, April. Unsupervised user behavior representation for fraud review detection with cold-start problem. *In Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 222-236). Springer, Cham.
4. **Li, Q.**, Wu, Q., Liu, X., 2019. Multi-scale and hierarchical embedding for polarity shift sensitive sentiment classification. *Lecture Notes in Computer Science(including subseries Lecture Notes in Artifical Intellifence and Lecture Notes in Bioinformatics)*, 11633LNCS, pp.227-238.
5. Da, Q., Cheng, J., **Li, Q.**, Zhao, W., 2019. Socially-attentive representation learning for cold-start fraud review detection. *Communications in Computer and Information Science*, 1069, pp.76-91.

Submitted Manuscripts:

6. **Li, Q.**, Wu, Q., Zhu, C., Zhang, J., Zhao, W., 2020. Inferable representation learning for cold-start fraud review detection. *Neurocomputing* (Accepted).
7. **Li, Q.**, Wu, Q., Zhu, C., Cao, L., Zhang, J., 2020. Deep Non-IID Paragraph Representation for Sentiment Analysis. *IEEE Access* (Under Review).

8. Zhu, C., Cao, L., and **Li, Q.**,2020. Unsupervised hierarchical and heterogeneous coupling learning on dynamic categorical data. *Journal of Machine Learning Research* (Under Review).

Others :

9. Zhao, W., **Li, Q.**, Zhu, C., Song, J., Liu, X. and Yin, J., 2018. Model-aware categorical data embedding: a data-driven approach. *Soft Computing*, 22, pp.3603-3619.
10. Zhu, C., Zhao, W.,**Li, Q.**, Li, P., Da, Q., 2019. Network embedding-based anomalous density searching for multi-group collaborative fraudsters detection in social media. *Computers, Materials and Continua*, 60(1), pp.317-333.
11. Xiang, L., Zhao, G., **Li, Q.**, Hao, W., Li, F., 2018. TUMK-ELM: A fast unsupervised heterogeneous data learning approach. *IEEE Access*, 6, pp.35305-35315.
12. Zheng, Y., Zhao, W., Sun, C., **Li, Q.**, 2019. Drug side-effect prediction using heterogeneous features and bipartite local models. *Computers, Materials and Continua*, 60(2), pp.481-496.
13. Cui, J., Long, J., Min, E., Liu, Q., **Li, Q.**, 2018. Comparative study of CNN and RNN for deep learning based intrusion detection system. *ICCCS 2018: Cloud Computing and Security (LNCS)*, 11067, pp. 159-170.

Table of Contents

List of Figures

Figure **Page**

List of Tables

CHAPTER 1

Introduction

1.1 Background

Sentiment analysis refers to the use of various techniques (e.g., text analysis, feature engineering, representation learning, and machine learning) to identify, extract, and quantify subjective information in source materials. It has brought great business values to both industries and individuals (Fan & Gordon 2014). For example, it helps a business to monitor the social opinion of a product and helps a customer to understand and select a service.

Sentiment analysis in social media is critical yet challenging because the source materials (i.e., reviews posted in social media) are with *high complexity*, *low quality*, and *uncertain credibility*. Specifically, words and sentences in a textual review may couple with each other (Teng et al. 2016, Jia et al. 2018), and they may have heterogeneous meanings under different contexts or in different language locales (Zhuang et al. 2017). These couplings and heterogeneities essentially determine the sentiment polarity of the review but are too complex to be captured and modeled (Ralaivola et al. 2010, Cao 2013). Also, social reviews contain a large number of informal words and typos (a.k.a., noise) but rare number of vocabularies (a.k.a., sparsity). As a result, most of the existing natural language processing (NLP) methods may fail to effectively represent social reviews. Furthermore, a large proportion of social reviews are posted by fraudsters (Mukherjee, Venkataraman, Liu & Glance 2013). These fraud reviews manipulate social opinion, and thus, they disturb sentiment analysis.

1.2 Current Work and Gap Analysis

Sentiment analysis methods can be classified into two paradigms: (1) hand-craft-feature-based methods (Turney 2002, Saif et al. 2016, Bravo-Marquez et al. 2014) and (2) representation-learning-based methods (Kim 2014, Passalis & Tefas 2017, Peters et al. 2018, Devlin et al. 2018). The former paradigm represents textual information as vectors by designated features (e.g., n-gram, bag-of-words, sentiment scores). In contrast, the later paradigm learns a model, such as neural networks and latent Dirichlet analysis, to automatically extract features from textual data. As shown in recent advance (Tang et al. 2014, Passalis & Tefas 2018, Vincent & Ogier 2019), representation-learning-based sentiment analysis methods achieve much higher accuracy compared to that based on hand-craft features, because the learned representation embeds many sentiment-related characteristics that cannot be captured by hand-craft features.

Although the existing representation-learning-based sentiment analysis methods demonstrated superior performance, most of them overlook the non-independent and non-identically distributed (non-IID) characteristics of complex textual data. Here, the non-IID characteristics include that (1) words and sentences couple with each other (non-independent; Teng et al. 2016, Jia et al. 2018), and (2) the same word and sentence may have different meanings under different contexts or in different language locales (non-identically distributed; Zhuang et al. 2017). The non-IID characteristics may hierarchically exist from word level to paragraph level; they determine the sentiment polarity of textual data but are hard to be represented. Recently, limited sentiment analysis methods begin to focus on the non-IID characteristics in their representation learning. For example, Teng et al. (2016) proposed a method to capture the nearby word couplings, Tang, Qin & Liu (2015) focused on the sequential relations of words, and Wang, Huang, Zhu & Zhao (2016) modeled the hierarchical heterogeneous meanings of words and sentences. However, these methods capture only partial of non-IID characteristics. As a result, they cannot comprehensively represent textual data with non-IID characteristics for sentiment analysis.

Furthermore, the current sentiment analysis methods mainly focus on standard articles but overlook the characteristics of social articles (e.g., social reviews, blogs, and social messages). In standard articles, texts are formal and follow specific syntaxes. On the contrary, texts in social articles always contain noise (e.g., typos, incomplete syntax, and irregular vocabularies) and are sparsity (e.g., the length of an article is short, and the number of vocabularies in an article is little). As a result, most of the existing sentiment analysis methods work well on the text in standard articles but may fail on the text in social articles. Advanced methods either reduce noise or alleviate sparsity for sentiment analysis on social reviews. Typically, the current noise reduction methods (Dey et al. 2016, De Boom et al. 2016, Wang, Wang, Zhang & Yan 2017, Arora et al. 2017*a*, Li, Wang, Zhang, Li, Chi & Ouyang 2018) first recognize noise by looking up pre-defined noise sets or adopting frequency-based detection models. Then, they reduce the impact of the recognized noise by re-weighting or ensemble strategies. However, two problems may arise in their recognition process: (1) pre-defined noise sets may not fully cover all noise, and (2) frequency-based detection models may fail when facing the sparsity. Consequently, their unrecognized noises may still damage sentiment analysis performance, even with re-weighting and ensemble strategies. Meanwhile, the current sparsity alleviation methods (Zuo et al. 2016, Liang et al. 2016, Lochter et al. 2016, Li, Li, Chi & Ouyang 2018) assume that a sparse short text is generated from a latent dense document, and try to insert words into the short text according to the latent document. However, the inserting quality cannot always be guaranteed because (1) many texts are independent (i.e., they cannot be generated from the same latent document), and (2) most of these methods are based on statistics, which may profoundly be affected by the noise in social reviews.

Moreover, most of the current sentiment analysis methods ignore the fact that the credibility of a social review is uncertain. They treat all reviews as honest when training the analysis model. As a result, their accuracy may be profoundly affected by fraud reviews, which occupies a large proportion of social reviews. Current fraud review detection methods mainly analyze user behavior and social relations (Ye & Akoglu 2015, Rayana & Akoglu 2015, Hooi et al. 2017, Liu et al. 2017). They assume a fraud review is posted by a user who has anomalous behavior (e.g., posting many reviews within a short period). They also assume collaborative fraudsters manipulate a group of reviews, which generates abnormal social relations between the fraudsters and their posted items. Because anomalous behavior and abnormal social relation have a good distinguishing ability between honest and fraud reviews, the existing methods have shown remarkable performance in fraud review detection (Rayana & Akoglu 2016). However, most of the existing fraud review detection methods may fail when facing the cold-start problem (i.e., a new user posts his/her first review). The new users in the cold-start problem bring two principal challenges below. (1) A new user does not have historical information for behavior analysis (You et al. 2018), which is required by most of the existing fraud review detection methods (Ye & Akoglu 2015, Rayana & Akoglu 2015). (2) A new user does not show any observed social relation, invalidating the detection of potential collaborative fraud review manipulation (Liu et al. 2017, Hooi et al. 2016).

In summary, the current sentiment analysis methods achieve significant performance but overlook the unique data characteristics in social media. Thus, they are not robust when they are applied on sentiment analysis for social articles which show high data complexity, low quality, and uncertain credibility. A systematically studying of the above characteristics is required to form a reliable sentiment analysis in social media, which expects to yield robust analysis results when facing high complex, low-quality, and incredible data.

1.3 Research Problems and Objectives

This research focuses on reliable sentiment analysis in social media. It systematically investigates the sentiment analysis techniques catering to the data characteristics of social media, which is summarized in Fig. 1.1.

Specifically, this research focuses on two research problems: *general sentiment analysis in social media* and *fraudulent sentiment analysis in social media*. The general sentiment analysis targets on tackling high data complexity and low-quality of social articles that are credible. The high data complexity typically exists in long texts (e.g., paragraphs), and the low-quality always accompanied by short texts (e.g., social reviews). Accordingly, this research split the general sentiment analysis problem into two subproblems: *paragraph sentiment analysis* and *short text sentiment analysis*. The fraudulent sentiment analysis handles the uncertain credibility issue, which is common and profoundly affects the precise sentiment analysis in social media. However, reliable labels for fraudulent sentiment analysis require a lot of artificial efforts that are scarce in most real applications. Therefore, this research first studies the *supervised fraudulent sentiment analysis* for effective fraudulent information modeling and then investigates the *unsupervised fraudulent sentiment analysis* for broader applications.

Modeling complex textual data for paragraph sentiment analysis is critical yet challenging. As discussed in Section 1.2, an essential complexity is non-IID characteristics (consist of couplings and heterogeneity), which cannot be comprehensively captured by the current methods. To fill this gap, this research objects to model word/sentence couplings and heterogeneity for better paragraph sentiment analysis. Furthermore, this research intends to model more effective sentiment-related features and integrates them with word/sentence couplings and heterogeneity to provide sufficient information for

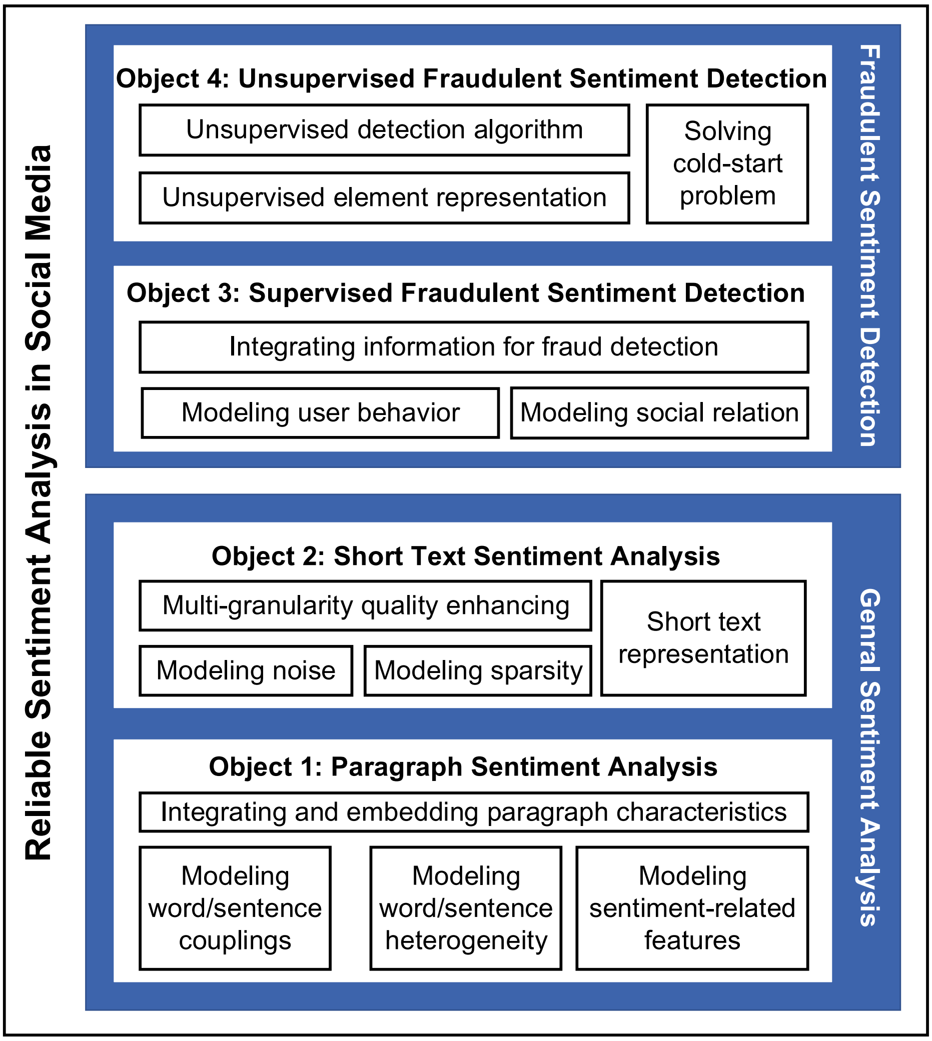

Reliable Sentiment Analysis in Social Media
Fraudulent Sentiment Detection
Object 4: Unsupervised Fraudulent Sentiment Detection
Unsupervised detection algorithm
Unsupervised element representation
Solving cold-start problem
Object 3: Supervised Fraudulent Sentiment Detection
Integrating information for fraud detection
Modeling user behavior
Modeling social relation
Genral Sentiment Analysis
Object 2: Short Text Sentiment Analysis
Multi-granularity quality enhancing
Modeling noise
Modeling sparsity
Short text representation
Object 1: Paragraph Sentiment Analysis
Integrating and embedding paragraph characteristics
Modeling word/sentence couplings
Modeling word/sentence heterogeneity
Modeling sentiment-related features

Figure 1.1: The research problems and their relations.

sentiment analysis in complex paragraphs. Accordingly, it has the following objectives for the paragraph sentiment analysis problem:

- modeling word/sentence couplings;
- modeling word/sentence heterogeneity;
- modeling sentiment-related features;
- integrating and embedding paragraph characteristics.

The challenges of short text sentiment analysis are mainly contributed by noise and sparsity. As shown in Section 1.2, none of the existing methods can effectively tackle the problems brought by both noise and sparsity. This research thus objects to model both noise and sparsity to enhance the quality issues of short text. Also, this research recognizes that the semantic meaning and the sentiment of a short text are determined by multi-granularity (e.g., characters, n-gram, and words). Consequently, it tries to enhance the quality of short text in multi-granularity. Lastly, this research aims to embed the enhanced short text into a vector representation space for representation-learning-based sentiment analysis. In summary, this research has the following objectives for the short text sentiment analysis problem:

- modeling noise;
- modeling sparsity;
- multi-granularity quality enhancing;
- short text representation.

To achieve supervised fraudulent sentiment analysis, this research leverages both user behavior and social relations. Although the existing methods have studied user behavior and social relations, most of them cannot effectively combine these two parts for comprehensive fraudulent sentiment analysis because these two parts reflect different information with various formats. Therefore, this research objects to model user behavior and social relations in a novel way and effectively integrated them for fraud detection. These objectives for the supervised fraudulent sentiment analysis problem are listed as follows:

- modeling user behavior;

- modeling social relation;
- integrating information for fraud detection.

Unsupervised fraudulent sentiment analysis requires discovering fraud information without the supervision of fraud labels. While mining the fraud information from social relations has been well-studied in unsupervised fashions, it is hard to model fraud-related user behavior without supervised labels because a method could not distinguish whether a behavior is related to fraud. This research objects to tackle this challenge by proposing an unsupervised element representation method to embed fraud-related user behavior. Furthermore, it aims to propose an unsupervised fraud detection algorithm based on its proposed unsupervised element representation. Meanwhile, this research also focuses on the cold-start problem. The fraudulent sentiment brought by the cold-start problem profoundly manipulates social opinions but hard to be detected, especially in unsupervised cases. This research thus has the following objectives for the unsupervised fraudulent sentiment analysis problem:

- unsupervised element representation;
- unsupervised detection algorithm;
- solving the cold-start problem.

1.4 Thesis Contributions

This thesis makes the following contributions:

- *Paragraph sentiment analysis:* This thesis studies the nature of paragraph polarity shift and proposes a multi-scale and hierarchical paragraph representation methods for polarity-shift-sensitive sentiment analysis. Furthermore, it studies the nature of non-IID characteristics and proposes a deep non-independent and identically distributed paragraph (non-IID) representation for implicit sentiment analysis. Extensive experiments on five large and two small real-world data sets demonstrate that the proposed method significantly enhances sentiment analysis in terms of both classification accuracy and sentiment representation quality through the comparison with four baseline methods and five state-of-the-art competitors with their eleven variants.

- *Short text sentiment analysis:* This thesis investigates the noise and sparsity problems in short text sentiment analysis. It proposes a *breaking-gathering* strategy and a corresponding neural network structure with an adversarial learning objective for short text sentiment analysis. Comprehensive experiments on five large real-world data sets demonstrate the proposed method significantly outperforms the state-of-the-art competitors.
- *Supervised fraudulent sentiment analysis:* This thesis studies the user individual behavior modeling and user social relations for fraudulent sentiment analysis. It proposes a supervised fraudulent sentiment analysis model by jointly considering user, item, review, and rating. Extensive experiments show that the proposed method effectively models user behavior and social relations, and it performs significantly better in detecting fraud reviews on four real-life social media data sets.
- *Unsupervised fraudulent sentiment analysis:* This thesis studies unsupervised fraudulent sentiment analysis with the cold-start problem. It proposes an unsupervised element representation, a social relation mining method, and an unsupervised fraud detection method, which suits for cold-start fraud detection. Extensive experiments show that the proposed unsupervised method (i) performs significantly better in detecting fraud reviews on four real online review data sets, and (ii) effectively infers new user representation in the cold-start problem with higher quality, compared to three state-of-the-art and two baseline competitors.

1.5 Thesis Organization

To build a theory system for reliable sentiment analysis in social media, this thesis explores three major challenges: high data complexity, low data quality, and uncertain credibility. It is organized by two parts: (1) The first part presents general sentiment analysis that address high data complexity and low data quality, including Chapter 3, 4 and 5; (2) The second part presents fraudulent sentiment analysis that address uncertain credibility, as in Chapter 6. The summary of each chapter is as follows:

- *Chapter 2:* This chapter presents a survey of reliable sentiment analysis. Specifically, it discusses the sentiment analysis paradigms, polarity-shift sentiment analysis, non-IID sentiment analysis, short text sentiment analysis, and fraudulent sentiment detection.

- *Chapter 3:* This chapter studies the polarity shift problem in paragraph sentiment analysis. It proposes a MUlti-Scale and Hierarchical representation method, MUSH, to learn a more accurate representation for polarity-shift-sensitive sentiment classification. The MUSH method adopts CNN with filters of different sizes to reveal multi-scale sentiment atoms and utilizes hierarchical multi-line CNN-RNN structures to jointlycapture polarity shift in both sentence and paragraph level.

- *Chapter 4:* This chapter focuses on modeling and capturing the complex couplings and heterogeneity in paragraph sentiment analysis. It comprehensively models couplings and heterogeneity by a novel representation framework. The framework is instantiated as a Multi-scale and hiErarchical DEep neural network with an Attention mechanism (MEDEA). The MEDEA method captures word/sentence couplings by a multi-scale convolutional-recurrent (CNN-RNN) structure and reveals the heterogeneous meanings of words/sentences in a paragraph by an attention mechanism. To regulate the representation and avoid over-fitting, MEDEA further hierarchically integrates these learned implicit features with explicit features, which are designed by sentiment priors.

- *Chapter 5:* This chapter investigates the problems caused by low data quality in short text sentiment analysis. Specifically, it tackles the noise and sparsity problems in short text sentiment analysis by learning multi-grain noise-tolerant patterns and then embedding the most significant patterns in a text as its representation. To achieve this goal, this chapter proposes a bi-level multi-scale masked CNN-RNN network to embed the most significant multi-grain noise-tolerant relations among words and characters in a text into a dense vector space.

- *Chapter 6:* This chapter studies fraudulent sentiment analysis in social media. It presents two novel methods for cold-start fraud review detection in supervised and unsupervised scenarios, respectively. The proposed methods embed user behavior and social relations of existing users in an inferable user-item-review-rating vector representation space. In this space, these methods can efficiently infer the most probable representation of a new user by a closed-form solution. Accordingly, they can effectively detect cold-start fraud reviews.

- *Chapter 7:* This chapter summarizes the thesis's content and contributions. It further discusses possible future avenues of research that can build to the work done in this study.

Chapter 2

Literature Review

2.1 Introduction

Sentiment analysis has a long history and broad applications. Typical sentiment analysis methods extract artificially designated features (e.g., bag-of-words, n-gram, and word frequency) from textual data and feed the extracted features into a classifier (e.g., naive Bayesian, decision tree, and random forest). Advanced sentiment analysis methods learn sophisticated sentiment-related features by models (e.g., deep neural networks) to achieve better sentiment analysis performance. However, most of the existing sentiment analysis methods assume textual data is formal and general. Accordingly, current methods may not be robust when analyzing sentiment in social media where textual data are informal with complex relations, and even fraud.

This chapter presents a survey of works that are related to sentiment analysis in social media. Specifically, it first discusses the general sentiment analysis methods, including paragraph sentiment analysis methods and short text sentiment analysis methods. Then, it further reviews the existing fraudulent sentiment analysis methods. This survey provides in-depth motivations of the research problems to achieve the research objectives and points out possible research avenues of reliable sentiment analysis in social media.

2.2 General Sentiment Analysis Methods

2.2.1 Hand-Craft Features for Sentiment Analysis

Typical sentiment analysis methods always explicitly extract or calculate hand-craft features from textual data to identify sentiment polarity. These hand-craft features are designed according to experts' understanding of the critical information for sentiment analysis. As a result, the sentiment analysis methods that adopt hand-craft features (e.g., word frequency, word or term presence, part-of-speech [POS] tagging, and sentiment lexicon) are effective and efficient.

For instance, the methods proposed by Ku et al. (2006) and Moghaddam & Ester (2010) involve word-frequency-based features, and the method proposed by Wiebe et al. (1999) builds features based on word or term presence to reflect paragraph sentiment. Other works (e.g., that in Liu & Zhang 2012) generate paragraph feature by picking up certain words (e.g., 'Goooooood') that rarely appear in a corpus. These special words often have a strong relation to sentiment polarity and have shown high effectiveness for sentiment classification (Liu & Zhang 2012). Another commonly used hand-craft feature is POS, which, for example, has been adopted by Xia & Zong (2011), Huang et al. (2017). The POS feature is generated by POS tagging, which marks up a word to a particular part of speech (e.g., noun, verb, adjective, or adverb) in a sentence. The POS feature involves a prior that the same word with different POS may have different meanings. As a result, it helps to recognize the heterogeneous meanings of a word. Furthermore, the methods proposed by Zhu et al. (2014), Qian et al. (2017) directly apply sentiment words as features, which assumes that sentiment words contribute mostly to the sentiment polarity of a paragraph. In addition, the method proposed by Bespalov et al. (2011) introduces the n-gram feature, which assumes that sentiment polarity also can be contributed by phrases.

2.2.2 Homogeneous Sentiment Analysis

Most of the current sentiment analysis methods assume a text has homogeneous sentiment. Advanced methods first capture and embed some kinds of information in a text into a vector representation space with the homogeneous sentiment assumption, and do the sentiment classification in such representation space. Therefore, representation learning is one of the most critical tasks in advanced sentiment analysis methods. According to the embedded information, the current representation learning methods for homogeneous sentiment analysis can be generally classified into four categories: (1) context-aware representation learning, (2) sequence-aware representation learning, (3) sentiment-aware representation learning, and (4) sentiment-polarity-shift-aware representation learning. These four categories are explained in detail in the following sections.

2.2.2.1 Context-Aware Representation Learning

Context-aware representation learning methods capture context information in textual data, where context information is the relationships between nearby words, sentences, and paragraphs. They embed this context information into a vector space to form context-aware deep features. As shown in recent evidence (Mikolov et al. 2013, Le & Mikolov 2014), these context-aware deep features reflect the semantic meaning of words, sentences, and paragraphs, and thus, they are effective for sentiment analysis.

To integrate the word-level contextual information, most of the existing methods build context-aware paragraph features based on context-aware word features (e.g., the word embedding generated by skip-gram and continue-bag-of-words). For example, the methods proposed by Chen (2017) and Arora et al. (2017*b*) represent a paragraph by summing the context-aware word features in the paragraph. Specifically, the method proposed by Chen (2017) uses a uniform weight summation based on customized word representations, and the method proposed by Arora et al. (2017*b*) learns a weighted summation for any word representations.

Many methods directly capture sentence- and paragraph-level context. They leverage different contextual information. For example, the method proposed by Le & Mikolov (2014) captures the contextual information of a paragraph by considering the interactions of words in the paragraph. Specifically, it requires the representation vectors of two paragraphs located closer in the representation space if these two paragraphs contain more number of similar words. Intrinsically, this method captures the contextual information that reflects the topic of a paragraph. Recent methods, such as the method proposed by Ren et al. (2016), explicitly incorporate the word-level context-aware representation into paragraph-level context-aware representation. These methods perform much better than other methods which consider only word-level or paragraph-level context. However, most of them ignore the different functions and meanings of a word at different positions because a word may have diverse sentiment polarities at different positions. To address this problem, the method proposed by Zhuang et al. (2017) further introduces a bag-of-discriminative-words representation, which captures different meanings of a word through word topic assignment.

Although context-aware representation learning methods have shown their strengths in modeling paragraph semantic meaning, they do not well capture the different sentiment polarities between words and paragraphs with similar context. For example, the words ‘good’ and ‘bad’ have different sentiment polarities but may have similar contexts (e.g., ‘good’ and ‘bad’ in ‘a good boy’ and ‘a bad boy’ share the same context: ‘a’ and ‘boy’), and thus, they may have similar vector representations generated by the existing context-aware representation learning methods. As a result, it is hard for a downstream sentiment classifier to distinguish the different sentiment polarities from the context-aware representation of ‘good’ and ‘bad’. To solve this problem, a method should carefully select a context to ensure words in the context will have the same sentiment polarity.

2.2.2.2 Sequence-Aware Representation Learning

Sequence-aware representation learning methods capture sequential relations in a paragraph to reveal sentiment polarity changes. They embed or integrate the captured sequential relations into a representation vector to form a sequence-aware representation for sentiment analysis.

A pioneering method proposed by Tang, Qin & Liu (2015) hierarchically captures sequential information from sentence-level to paragraph-level. At sentence-level, this method captures and embeds the sequential information of words in each sentence into a vector by convolution neural networks or long-short-term memory (LSTM) networks. At paragraph-level, it further embeds the sequential information of sentences into a paragraph representation vector. Although this method shows advances in hierarchical sequential information embedding, it may fail to precisely represent sentiment polarity when the sentiment polarities between a word and n-gram are inconsistent, because it splits a sentence as sequences of individual words. For example, ‘not bad’ has a positive or neutral sentiment polarity, but the split word ‘bad’ has a negative sentiment polarity. For another example, ‘a great deal of’ has a neutral sentiment polarity, which is different from that of the individual word ‘great.’ To address this issue, the method proposed by Tang, Qin, Wei, Dong, Liu & Zhou (2015) further introduces an end-to-end paragraph-segmentation framework for paragraph sentiment analysis. Instead of dividing a paragraph into sequences of individual words, this method segments a paragraph into several phrases and predicts the sentiment polarity of the paragraph based on the sentiment polarities of the segmented phrases.

The above methods for sequence-aware representation are all based on recurrent neural network (RNN) with different structures. The RNN structure profoundly determines the sentiment analysis performance. Recently, many efforts have been made on improving RNN structures to achieve better paragraph representation performance for sentiment analysis. For example, a method revised the vanilla RNN as a batch-normalized RNN to fix the vanishing gradient problem in the RNN training process (Margarit & Subramaniam 2016). This method not only dramatically improves the training speed of the representation process but also significantly increases the sentiment classification accuracy. Another method deals with the long-term dependency vanishing problems caused by RNN (Lin et al. 2017). This method records the output states at each RNN step and feeds all of the previous outputs to the next RNN step by introducing a self-attentive mechanism that assigns a weight to each state for a better sentence representation.

2.2.2.3 Sentiment-Aware Representation Learning

Sentiment-aware representation learning methods directly learn sentiment-related features in a data-driven fashion. The features learned by sentiment-aware representation methods are more sensitive to a specific sentiment analysis task compare to that learned by context-aware and sequence-aware representation learning methods. Typically, sentiment-aware representation learning methods require to work with other representation learning methods to enhance the generalization ability of the learned features.

A preliminary method for sentiment-aware representation learning is proposed by Maas et al. (2011). This method uses a probabilistic model to capture the contextual and sentiment information in a paragraph jointly. It learns word representation vectors by maximizing the posterior probability of the paragraph sentiment polarity, and extensive experimental results have demonstrated its advances. Following this work, the method proposed by Tang et al. (2016) jointly embeds paragraph-level sentiment label and word-level contextual information into a representation vector to capture both sentiment and contextual information. In contrast, the method proposed by Vo & Zhang (2015) first embeds the sentiment information of a paragraph into a paragraph representation vector. It then leverages the sentiment lexicon information in the paragraph to provide a direct relationship between the paragraph and its sentiment polarity.

2.2.2.4 Sentiment-Polarity-Shift Representation Learning

Polarity-shift refers to that the sentiment polarity is changing among a paragraph. Polarity-shift significantly affects the sentiment of a paragraph, but it is hard to be detected. Accordingly, effectively capturing and embedding polarity-shift in a paragraph is critical for sentiment analysis.

The existing polarity-shift-aware paragraph embedding methods can be generally classified into three categories. The first category adopts a term-counting approach (Asghar et al. 2017), which represents a paragraph according to the total number of positive and negative sentiment terms contained in the paragraph. The methods in this category cannot accurately detect the sentiment polarity of a word if the word is near a word that will cause polarity-shift. The second category uses a machine-learning-based approach to recast a polarity-shift detection problem as a statistical classification task (Li et al. 2010, Ikeda et al. 2010). The methods in this category first use a polarity-shift classifier to detect the polarity-shift locations in a paragraph and then splits the paragraph into non-polarity-shift parts and polarity-shift parts. After that, they train different sentiment classifiers in different parts to alleviate the polarity-shift affections. The third category combines term-counting and machine-learning-based methods and shows superior performance (Kennedy & Inkpen 2006, Xia et al. 2016).

2.2.3 Heterogeneous Sentiment Analysis

Heterogeneous sentiment analysis methods have the hypothesis that the same text or content may appear different sentiment in different contexts or be posted by different subjects. They assume that hierarchical heterogeneity of text sentiment exists in social media. Different methods focus the heterogeneity at different levels. Accordingly, the existing heterogeneous sentiment analysis methods can be classified into text-level heterogeneous sentiment analysis, human-level heterogeneous sentiment analysis, and domain-level heterogeneous sentiment analysis.

2.2.3.1 Text-level Heterogeneous Sentiment Analysis

The text-level sentiment heterogeneity has three-folds of meaning: (1) different parts of a text may have different sentiment polarities, (2) a word or a sentence may have different sentiment polarities in different positions in a text, and (3) different words and sentences have different contribution to the sentiment polarity of text. The text-level heterogeneous sentiment analysis methods intend to capture these kinds of heterogeneity.

To reveal the inconsistent sentiment polarities of different parts in text, Tang, Qin, Wei, Dong, Liu & Zhou (2015) proposed a joint segmentation and classification framework for paragraph sentiment analysis. In the first stage, this method segments paragraph into several phrases and analyzes the sentiment polarities of these phrases. In the second stage, it predicts the paragraph sentiment based on the sentiment polarities of all phrases. Specifically, several candidate phrases are generated by a beam search based generation model. The segment quality of these candidate phrases are then evaluated by a segmentation ranking model. The phrases with top-K ranking values are selected to make sentiment polarity classification for the paragraph in a voting procedure. The goodness of this method is that it captures phrase-level sentiment, which is more accurate than the word-level sentiment in a paragraph and considers the sentiment heterogeneity. However, the method ignores the sequential information within and between phrases, which reduces the phrase-level sentiment classification accuracy and limits the ability to capture the phrase-level sentiment changing.

To capture the heterogeneous sentiment polarities of a word in different positions, Zhuang et al. (2017) used a topic model to assign different topics for a word and proposed a bag-of-discriminative-words representation method. This method assumes that different word has different discriminative power on different topics. In this method, the word-topic pairs with large discriminative power are selected through a discriminatively objective-subjective latent Dirichlet allocation in a supervised fashion. The experimental results demonstrate that this method outperforms its competitors. Intrinsically, the superiority of this method is contributed by its ability to capture the heterogeneous meaning of a word through a topic assignment. However, this method is lack of efficiency when embedding a large number of paragraphs because of using Gibbs sampling, which is time-consuming, to infer the topic assignment. Moreover, when embedding a new paragraph, the inference process should be carried out again, which also costs lots of time.

Similar to the work of Zhuang et al. (2017), Hai et al. (2017) proposed a probability graphic model to assign the polarity for each word. This method assumes a word may have different sentiment polarity in different aspects. Thus, it also assigns an aspect label for each word to capture this heterogeneous sentiment distribution. Since this method has a more sophisticated design to model the sentiment generation in a paragraph, it shows significant performance improvement in the experiments. However, it treats every word equally to make the same contribution to the final sentiment polarity of a paragraph and disregards the order information of words due to a bag-of-words-based input. Another weakness of this method is that the number of aspects and the number of aspect-level sentiment polarities should be manually assigned, which requires additional expert knowledge or side information.

To tackle the heterogeneous contribution from different words, Lin et al. (2017) proposed a novel RNN structure based on the self-attentive mechanism for sentiment analysis. In this work, the model records the output states in RNN each step, and feed all of them into the next stage. Meanwhile, the importance of each output state is learned by considering its relation to text sentiment polarity and its link to other output states. Intuitively, this method enhances the contribution from words that are highly related to the sentiment polarity of text. Empirical results demonstrate the effectiveness of considering such heterogeneous contributions. However, this method overlooks the text-level heterogeneity that a word may have different sentiment polarities in different positions in a text.

2.2.3.2 Human-level Heterogeneous Sentiment Analysis

Human-level heterogeneity is caused by the personalized sentiment behavior of users in social media. The same text posted by several users may involve different sentiment. Different from text-level heterogeneity, human-level heterogeneity is not reflected by the text itself. Human-level heterogeneous sentiment analysis methods leverage the social characteristics of users to enhance the analysis performance.

Jiang et al. (2011) focused on distinguishing the sentiment polarity of short text posted in social media. For short text, the sentiment polarity is always ambiguous. Jiang et al. (2011) enhanced the presentation of a short text by introducing (1) the other text posted by the same user and (2) the replies of the text posted by other users. Intrinsically, this approach adds additional text information to the short text guided by social information. The experimental results show this approach dramatically improves the performance of sentiment analysis. Although this method considers the social information of a user, it is only used to complement the text. The sentiment analysis results are still based on text mining, which loses the text-independent sentiment information.

Different from Jiang et al. (2011), Tan et al. (2011) used social information, i.e. homophily in social network, as the complement of text information for sentiment analysis. The basic idea behind this work is that users that are linked may be more likely to hold a similar sentiment. By forcing the linked user to share a similar sentiment, this work captures the group sentiment preference of users. Compared with textual feature-based methods, this work leads to a statistically significant improvement in terms of sentiment classification. The strength of this work is using inter-user relationships to reflect human-level heterogeneity, which inducing performance lift. The weakness of it is that this work ignores the user's behavior that involves the text posted by the same user in the past.

To capture the behavior and social relations of a user simultaneously in sentiment analysis, Hu et al. (2013) proposed a method to incorporate these two kinds of information into supervised sentiment analysis. Specifically, it transforms user-message matrix $\mathbf{U}$ and user-user matrix $\mathbf{F}$ to sentiment consistency matrix $\mathbf{A}_{sc} = \mathbf{U}^\top \times \mathbf{U}$ and emotional contagion matrix $\mathbf{A}_{ec} = \mathbf{U}^\top \times \mathbf{F} \times \mathbf{U}$. The sentiment consistency matrix reflects user behavior. $\mathbf{A}_{sc_{ij}} = 1$ indicates that the i-th and j-th messages are posted by the same user, and sentiments of the two messages are similar. The emotional contagion matrix reflects the user-user similarity. $\mathbf{A}_{ec_{ij}}$ indicates that the author of the i-th message is a friend of the author who wrote the j-th message, and sentiments of the two messages are similar. The method combines $\mathbf{A}_{sc}$ and $\mathbf{A}_{ec}$ to calculate the Laplacian matrix that used in sentiment learning objective function. Instead of jointly incorporating behavior and social relations, this work also identified the social theories that exist in microblogging data, which build a foundation for the following work on this topic. It should be noted that this method only captures second-order relationships. For the higher-order relationships (e.g., the influence from the friends of a friend), this method does not reveal.

For higher-order relations, Tang, Nobata, Dong, Chang & Liu (2015) proposed a propagation-based sentiment analysis method. This method using the propagation process for sentiment analysis, which is intrinsically different from previously discussed classification-based methods. It assumes the sentiment in social media can propagate from one user to another and finally reflected in the posted text. To do the propagation, this method first builds a graph/network that consists of three components. The first is a microblog-microblog network $\mathbf{R}^{tt}$ based on social theories where $\mathbf{R}^{tt}_{ij} = 1$ indicates the sentiment polarity of the i-th and j-th microblog may be correlated. The second is a word-word network $\mathbf{R}^{ww}$ where $\mathbf{R}^{ww}_{ij} = 1$ indicates the i-th and j-th word may be correlated. The last is the microblog-word bipartite graph $\mathbf{R}^{tw}$ where $\mathbf{R}^{tw}_{ij}$ denotes the frequency of the j-th word in the i-th microblog. After building the network, the method does the propagation based on the links and weights of the network. Through this approach, they captured higher-order relations between users and words. In addition, the method has another advantage that it can handle both labeled and unlabeled data, which largely reduces the cost of data annotation in the pre-processing stage. Despite lots of advantages, this method needs a large amount of data for a robust analysis result. It does not suit for small data analysis due to its basic assumption about sentiment propagation.

2.2.4 Domain-level Heterogeneous Sentiment Analysis

The domain-level heterogeneity refers to the differences of sentiment in different domains, which is widespread in social media. This heterogeneity is caused by some unique biases and different backgrounds in each domain. Treat the sentiment in different domain homogeneously will cause obvious inaccurate results. Although the difference exists, consensus sentiment information is also shared by different domains. Leveraging this common sentiment information will improve the analysis performance for the domains with rare studying samples. The domain-level heterogeneous sentiment analysis methods hold the hypothesis of such heterogeneity and intend to figure out the common and unique cross-domain sentiment information. Current methods mainly make efforts through three ways: (1) transforming a domain to another; (2) mapping heterogeneous domains to a homogeneous space; and (3) distributed using intact information in different domains.

The method proposed by Pan et al. (2010) uses a labeled domain as training data for sentiment analysis in other domains. The key idea is to adopt a clustering method to align the words in different domains, which uses domain-independent words as a bridge. It equals mapping words in an unlabeled domain to the words in the labeled domain. In this way, only embedding in the labeled domain needs to be trained for sentiment analysis. The pro of this method is that it links different domains by the word-level similarity. However, the cons of it include: (1) clustering based on domain-independent words is too weak to align different domains well; (2) embedding information is only leveraged from a labeled domain, which causes losing richer information in unlabeled domains; and (3) the text representation via this method is sentiment independent.

Instead of mapping a domain to another, the Liu et al. (2015) tried to extract common features from different domains. They propose a semi-supervised method to combine common and specific features of domains for topic-adaptive sentiment classification.

Similarly, the method proposed by Ganin et al. (2016) embeds different domains into a common space, in which a sentiment classifier can be learned. In this work, neural networks are learned as the functions to map different domains to a common space, and the learning procedure is with an adversarial training strategy. The key point of this idea is to learn the common space and sentiment classifier jointly. Specifically, two classifiers with different tasks are trained simultaneously. A classifier is used to classify the sentiment polarity, while another classifier is trained to distinguish domains. The overall objective of this method is to minimize the loss value of the former classifier and maximize the loss value of the later one. In this way, the representation of text in the common space is friendly to sentiment analysis, yet it is hard to identify domain information. However, this method does not capture the sequential information of a paragraph; thus, it is not sensitive to the polarity shift in text.

The method proposed by Fernández et al. (2016) has a similar idea to the method proposed by Ganin et al. (2016). It also tries to map different domains into a common space. However, this method does not couple the mapping function learning and sentiment classifier learning together. Instead, it explicitly learns the mapping function according to the heterogeneous distributions in different domains, and it pays more attention to the highly predictive term that behaves similarly in different domains. This method reveals the consensus information about distributions in different domains. However, it assumes possible distributions of different domains. If real data does not fit any assumed distributions, this method cannot leverage the sentiment information.

As can be seen, the above work either maps a domain to another or learns a common space for all domains. However, both of these two kinds of method lose the intact information for sentiment analysis. Specifically, the former loses the unique information in the target domain, and the latter only remains a small piece of common subspace in all domains. To make full use of information in different domains, some current methods, such as the work of Wu & Huang (2016) and Wu et al. (2017), attempt to use a distributed way to represent the cross-domain information. Typically, this kind of method learns a similarity between the target domain and source domain and selectively applies the source domain information according to its similarity to the target domain. The key difference between each method is the way to learn the similarity between domains. Wu & Huang (2016) constructs sentiment polarity relation graphs for each domain from syntactic parsing results and learns the domain similarity according to the similarity of the graphs. Wu et al. (2017) designs textual content-based domain similarity and sentiment expression-based domain similarity. Both of these methods capture domain relations in different views and leverage the information in all domains for sentiment analysis in the target domain. However, how to determine the view that should be used to measure domain similarity and how to integrate domain similarity from different views are still open problems.

Another method that comprehensively captured cross-domain information is proposed by Bollegala et al. (2016). This method constructs three objective functions to leverage (1) distributional properties of the common embedded space; (2) sentiment label constraints in the source domain paragraph; and (3) geometric properties in the unlabeled paragraph in all source and target domains. The experimental results demonstrate the superior performance of this domain-level heterogeneous sentiment analysis method. However, a shortage of this method is that doing the trade-off of importance is hard for its three objectives. Bollegala et al. (2016) did not theoretically analyze and give a guide for the weight setting for these three objectives. The bias for any part may have a significant impact on the sentiment analysis result.

2.2.5 Short Text Sentiment Analysis

Recently, short text sentiment analysis has been attracting lots of attention because most of the textual data on social media are short text. Sentiment analysis on short text has more significant challenges than that on formal text. These challenges are mainly caused by two characteristics of short text: *noise* and *sparsity*, where *noise* infers to informal words and typos, and *sparsity* means the rare number of vocabularies in a text (because of the short length limitation). On the one hand, *noise* prevents the word representation. On the other hand, *sparsity* reduces text representation quality. Accordingly, sentiment analysis methods are hard to capture sufficient high-quality information from a short text to detect its sentiment polarity.

To tackle the noise problem in short text representation, the most widely used method is filtering noises by looking up pre-defined noise sets. However, pre-defined noise sets are fixed and cannot comprehensively cover noises. Recently, some advanced learning methods have been proposed for the noise problem. For example, Dey et al. (2016) propose a feature selection method to select noise-tolerant features from a set of designated lexical, syntactic, semantic, and pragmatic features. These features include character-level gram features, word-level gram features, part-of-speech (POS), named entity features, word overlap features, phrase overlap features, subjectivity/objectivity agreement feature, and close attachment of negations. De Boom et al. (2016) propose a method that weighted sums word embeddings for a short text representation and adopts a median-based loss function to reduce the effect of noises in the weight learning process. However, the summation of word embeddings ignores the ordinal of the words, which may also determine the semantic meaning of the short text. Catering to the topic model, Li, Wang, Zhang, Li, Chi & Ouyang (2018) model the noises in a short text by a common distribution to filter out the noise influences. However, this method may equally treat the noises that have different meanings.

2.3 Fraudulent Sentiment Analysis Methods

Fraudulent sentiment analysis is a task that intends to distinguish fake or spam opinion. It is critical in social media sentiment analysis because a large proportion of opinions in social media is unreliable. The existing methods leverage different information for fraudulent sentiment analysis in social media. According to the leveraged information, these methods can be categorized into three groups: (1) text-content-based method, (2) rating-distribution-based method, and (3) social-relation-based method. Recently, fraudulent sentiment analysis methods begin to focus on another challenging task that is to detect fake opinions when facing the cold-start problem (i.e., a new user post his/her first review). Effectively detecting cold-start fraudulent sentiment prevents opinion manipulation at the very beginning stage and thus can primarily reduce fraudulent sentiment in social media.

2.3.1 Text-Content-Based Fraudulent Sentiment Analysis

Text-content-based fraudulent sentiment analysis methods determine fraud opinions from textual features. These methods either introduce designated fraud-sensitive features for fraudulent sentiment detection or propose fraud-oriented sentiment analysis procedures.

Jindal & Liu (2008) pioneer text-content-based fraudulent sentiment analysis. They leverage duplicated and nearly duplicated information in social media to identify fraudulent sentiment. Specifically, they identify the following reviews as fraud reviews: (1) duplicated reviews from different users on the same product, (2) duplicated reviews from the same user on different products, and (3) duplicated reviews from different users on different products. Furthermore, they use the identified reviews that contain fraudulent sentiment as training samples to train a fraudulent sentiment classifier, and then, they check the other reviews by the trained classifier. Although duplicated reviews are with high probability to contain fraudulent sentiment, they do not cover all fraud reviews (e.g., the review wrote by a user but with spam opinion). As a result, the method proposed by Jindal & Liu (2008) achieves high precision but low recall.

Benevenuto et al. (2010) intensively analyzed Twitter samples to discover textual features other than duplicated reviews for fraudulent sentiment analysis. Firstly, they manually classified a set of users into spammers and non-spammers according to their observation and understanding, and they assign the reviews posted by the spammers as fraud reviews that contain fake sentiment opinion. Then, they designed many textual features to studied the relations between these features and the assigned fraud reviews. Meanwhile, they investigated the relations between user attributes (e.g., age, gender, and occupation) and the assigned fraud reviews. In the above analysis, they found some patterns that exist in the text content of fraud reviews. For example, fraud reviews contain a much higher fraction of URLs compared to the other reviews. Lastly, they transformed the fraud-related patterns to features and demonstrated these features could enable a better fraudulent sentiment analysis compared to the methods that only consider duplicated reviews. However, these features are all designed by empirical observation in a single domain (i.e., Twitter). Consequently, the generalisability of these features is weak and may not fit fraudulent sentiment analysis in other domains.

Feng et al. (2012) introduced general features to tackle the generalisability problem. These general features include word features (e.g., bag-of-words), sallow syntax features (e.g., POS), and deep syntax features generated by probabilistic context-free grammar (CFG) parse trees. Specifically, the probabilistic CFG parse trees generate deep syntax features following four rules: (1) unlexicalized production rules, (2) lexicalized production rules, (3) unlexicalized production rules combined with the grandparent node, and (4) lexicalized production rules (i.e., all production rules) combined with the grandparent node. Feng et al. (2012) evaluated the performance of these features on data sets from different domains. The evaluation results demonstrate that the introduced general features consistently improve fraudulent sentiment analysis performance in all domains.

Instead of designing fraud-sensitive features, Ott et al. (2011) proposed a novel learning procedure catering to the fraudulent sentiment analysis. This method divides a fraudulent sentiment analysis into three subtasks. In the first subtask, this method categorizes text content as deceptive or truthful by an n-gram-based classifier. In the second subtask, it detects psycholinguistic deception, assuming deceptive statements would exemplify the psychological effects of lying (e.g., negative emotion increasing and psychological distancing). In the last task, this method identifies the genre of text content. Specifically, it treats deceptive and truthful writing as sub-genres of imaginative and informative writing, respectively. The above three tasks complement each other for fraudulent sentiment analysis. By jointly performing these tasks, this method can comprehensively detect fraudulent sentiment.

Although text-content-based methods capture the fraud information reflected by text content, they ignore other more abundant information (e.g., user behavior and rating outliers) for fraudulent sentiment analysis. Furthermore, most of these methods have a high computational cost to extract textual features and classify fraud texts. As a result, they may not be suitable for real-time applications with a large amount of data. The last but not the least, text-content-based methods cannot distinguish fraudulent sentiment of the same text content for different products. For example, a positive review of an excellent product is trustful. However, the same review of an inferior product may be fake. Consequently, methods based only on text content cannot identify whether the sentiment of this review is fraud or not. To tackle the above problems, more recent work focuses on other social media information other than text content for fraudulent sentiment analysis.

2.3.2 Rating-Distribution-Based Fraudulent Sentiment Analysis

The existing rating-distribution-based fraudulent sentiment analysis approaches leverage the reviewing behavior reflected by rating distributions to identify fraudulent sentiment. Compared with the text-content-based method, the rating-distribution-based method is more efficient and involves more abundant information from user behavior.

The method proposed by Lim et al. (2010) detects spammers according to the following patterns: (1) a group of spammers usually manipulate the opinions of specific products collaboratively, and (2) their rating distributions always deviated from honest reviewers. This method adopts a scoring strategy to evaluate the suspicious degree of each reviewer. Under the scoring strategy, this method assigns very similar scores to spammers, which have very different scores from that of honest reviews. Lim et al. (2010) fed the reviews that have a high suspicious degree into a commercial spammer detection software to verify the effectiveness of the proposed suspicious spammer scoring strategy. The experimental results demonstrate that this scoring strategy is precise in identifying spammers. Furthermore, adopting this scoring strategy avoids a high computational cost because this strategy does not need to extract features from natural review text. However, this method overlooks textual information, which may also contribute to fraudulent sentiment analysis.

The method proposed by Jindal et al. (2010) detects fraudulent sentiment by identifying unusual rating patterns. It defines several expectation rules according to rating distribution and formulates the unusual rating pattern identification problem as an unexpected rule detection task. This method is generalizable in all domains because it depends only on domain-independent rating patterns.

Recent rating-distribution-based fraudulent sentiment analysis approaches assume that fraudsters post fake reviews in a short time to increase or damage the popularity of their target product rapidly. In other words, they assume fraudulent sentiment posting is always bursty and may only occur in specific time intervals. Under this assumption, these methods build different models to detect and analyze fraudulent sentiment. For example, the method proposed by Xie et al. (2012) detects the time windows that contain fraudulent sentiment posting activities by a hierarchical scanning strategy: it first identifies a large time window to smooth possible noise changes along with time series, and it then scales down the window size to quickly locate the suspicious reviews. For another example, the method proposed by Günnemann et al. (2014) detects the time intervals where abnormal rating behaviors may occur.

2.3.3 Social-Relation-Based Fraudulent Sentiment Analysis

Social-relation-based fraudulent sentiment analysis methods utilize relational information between users and products in social media to identify fraud opinion. Typically, they create a graph structure to abstract relational information. In the graph structure, nodes can represent users and products, and links can indicate relations introduced by reviews and ratings. Compared with text-content-based and rating-distribution-based methods, social-relation-based methods capture more comprehensive information for a more accurate fraudulent sentiment analysis.

The method proposed by Jiang et al. (2014) automatically spots fraud users based on connectivity patterns of nodes in a directed user-product graph. Specifically, it first extracts five features for each node: (1) out-degree and in-degree, (2) hubness and authoritativeness score, (3) betweenness centrality, (4) in-weight and out-weight (if the graph is a weighted graph), and (5) values in the singular vector corresponding to a node. It then calculates a normality value for each feature and concatenates them to form a feature space. Lastly, it adopts a distance-based method on the feature space to spot fraud users.

Hooi et al. (2016) proposed another method to detect fraud opinions based on a user-product graph. In contrast to the distance (or relative positions) used by Jiang et al. (2014), this method focuses on the dense subgraph (a subgraph that has a much higher link density compared with other subgraphs) in a user-product graph by assuming fraudsters would collaborate to manipulate social opinions. By adopting a designated density evaluation metric, this method can effectively detect fraud opinions. Hooi et al. (2016) also demonstrated that their proposed method could effectively and efficiently detect fraudulent sentiment even a review is with three types of camouflage: random camouflage, biased camouflage, and camouflage by hijacked accounts.

The above discussed social-relation-based methods leverage only relational information yet ignore other information (e.g., text content and rating) in social media. As a result, they may have a high detection precision but a low detection coverage rate. To fill this gap, Rayana & Akoglu (2015) bridged relational information and other meta-information in social media by a unified framework. This unified framework models relational information as a user-review-product network, where each review is a link that connects the nodes of its corresponding user and product. It also embeds other meta-information into a feature vector for all users, reviews, and products. After the modeling and embedding, this framework identifies fraudulent sentiment by an unsupervised learning technique that integrates the network and feature vector.

Wang, Liu, He & Zhao (2016) made a further effort by learning a vector representation that jointly embeds relational information and text content. Initially, the method represents relational information in terms of eleven factors, such as time, locations, social relations, contact information. Meanwhile, it represents text content by the bi-gram feature. Then, the method collectively learn the final vector representation from the initial representation of both relational information and text content. Lastly, the method can precisely detect fraudulent sentiment based on the final vector representation.

2.3.4 Cold-Start Fraudulent Sentiment Analysis

Although recent years have seen significant progress made in fraudulent sentiment analysis, detecting fraud reviews with the cold-start problem is still a very challenging task and has rarely been studied. Specifically, the new users in the cold-start problem pose the two significant challenges below. (i) A new user does not have historical information (You et al. 2018). However, most of existing fraudulent sentiment analysis methods require the historical information to analyze user behavior (Ye & Akoglu 2015, Rayana & Akoglu 2015). (ii) A new user does not show any clear social relation, which invalids the detection of potential collaborative fraud review manipulation (Hooi et al. 2016, Liu et al. 2017).

Limited efforts have been paid on the cold-start problem in fraudulent sentiment analysis. The primary solutions for cold-start fraudulent sentiment analysis are review content-based methods, such as (Lim et al. 2010, Li et al. 2011, Ott et al. 2011, Mukherjee, Venkataraman, Liu & Glance 2013, Xu et al. 2013, Li et al. 2015, Hovy 2016). These methods identify spam patterns, such as outlier review length and the large percentage of capital words, in the review content. Consequently, they avoid the adverse effects brought by the lack of historical information and social relations in the cold-start case. However, the above methods focus only on the review content but overlook fraud-related information, such as rating distribution and social relation.

Recently, Wang, Liu & Zhao (2017) embeds the relation between existing users, items, and reviews into the review representation to capture information in addition to review content. To achieve that, they assume a user's behavior will determine the review content written by the user for an item. Subsequently, they build a learning-objective function that the representation vector of a user plus the representation vector of an item should equal to the representation vector of the review written by the user for the item. Then, they obtain the review representation vector that embeds the relation between existing users, items, and reviews by solving the learning-objective function. Finally, they feed the review representation vector into a support vector machine (SVM; Cortes & Vapnik 1995) to analyze fraudulent sentiment. This process makes significant progress in cold-start fraudulent sentiment analysis. Following Wang, Liu & Zhao (2017), You et al. (2018) further leverage domain knowledge and the attribute information of users and items to achieve a more informative representation. They demonstrate that their proposed method can achieve state-of-the-art performance when attribute information is available, especially when items are from different domains.

Although the above method can alleviate the cold-start problem, they are insufficient when dealing with real-life fraud reviews (Mukherjee, Venkataraman, Liu & Glance 2013). The rationale is that all of them determine fraudulent sentiment based on only review representation. As a result, they may fail to distinguish fraudulent reviews from honest reviews when these reviews have the same content, as analyzed in Chapter 2.3.1.

Chapter 3

Polarity-Shift Sentiment Analysis

3.1 Introduction

Sentiment analysis aims to determine the polarity of a given text. As discussed in Section 2.2, one of the most popular and efficient ways for sentiment analysis is the machine-learning-based approach. This kind of approach not only can significantly reduce the hand-work cost of building sentiment lexicon but also can capture more complex representation structures compared with that based on lexicon rules. A fundamental task of the machine-learning-based analysis is to represent a paragraph as a vector in a suitable vector space, in which a classifier can effectively analyze sentiment polarity.

Currently, lots of paragraph representation methods for sentiment analysis have been proposed to reveal complex information in a paragraph. These relationships are finally embedded into a vector space to represent the meaning of the paragraph for sentiment analysis. Although a variety of relationships have been studied, three kinds of relationships attracted most of the focus. (1) The context information (context refers to the content surrounded a word, sentence, and paragraph): several context-aware representation methods have been proposed, such as the methods proposed by Le & Mikolov (2014), Ren et al. (2016), Chen (2017), Arora et al. (2017*b*), Zhuang et al. (2017). (2) The sequential relationship, which contains in the order of word and sentence in a paragraph: the sequential-aware method, including these proposed by Tang, Qin & Liu

(2015), Tang, Qin, Wei, Dong, Liu & Zhou (2015), Margarit & Subramaniam (2016), Lin et al. (2017), reveals the sequential relationships that can capture different meanings of a word and sentence in different places. (3) The relationship between a paragraph and sentiment, which is the most direct relationship for the sentiment analysis task: the representative sentiment-aware methods include these proposed by Maas et al. (2011), Vo & Zhang (2015), Xia et al. (2015), Tang et al. (2016), Hai et al. (2017).

Although significant progress has been made in paragraph representation, most of the current methods overlook *sentiment-polarity-shift*, which means the sentiment polarity is slightly or extremely changed by some indicator words or expressions. Sentiment-polarity-shift is very important for sentiment classification, but it is hard to be captured. For example, negative words, such as *not*, can change the sentiment polarity of words, such as *happy*, from positive to negative. Similarly, the turning words, such as *however*, will cause a sentiment polarity turnover. Obviously, a representation method that ignores sentiment-polarity-shift may deviate the sentiment polarity of a paragraph in its representation. A typical example of this method is the bag-of-words representation, which presents a paragraph as the frequency of each word. The bag-of-words representation provides the proportion and frequency of emotional words for a downstream sentiment classifier. However, when the negative words (e.g., *not*) or turning words (e.g., *however*) exist, the bag-of-words representation cannot indicate the position of such words. As a result, it enabled classifier may fail to distinguish different sentiment polarity.

Recently, limited paragraph representation methods have been proposed for polarity-shift-sensitive sentiment classification. These limited efforts for polarity-shift-sensitive paragraph representation belong to three paradigms: (1) counting sentiment words with reversing human-labeled polarity-shifted words, (e.g., the method proposed by Asghar et al. 2017); (2) splitting a paragraph by machine-learning-determined polarity-shifted words, and then, embedding the separated paragraph (e.g., the methods proposed by Li et al. 2010 and Ikeda et al. 2010); (3) the combination of the previous two approaches (e.g., the methods proposed by Kennedy & Inkpen 2006 and Xia et al. 2016). However, these methods may fail when facing the following challenges: (1) polarity-shift is with multiple forms; and (2) polarity-shift exists in both sentence- and paragraph-level with complex long-term dependence. For the first challenge, polarity-shift may be caused by multiple forms, including the explicit negative words, explicit turning words, implicit tone, and implicit context. It is hard for the existing methods to capture these explicit and implicit polarity-shift factors simultaneously. For the second challenge, the existing polarity-shift-sensitive paragraph representation methods do not model the hierarchical and sequential structure of a paragraph. Consequently, they are difficult to embed polarity-shift correctly.

To address the above issues, this chapter studies the nature of paragraph polarity-shift and proposes a **mu**lti-**s**cale and **h**ierarchical paragraph representation method (MUSH) for polarity-shift-sensitive sentiment analysis. The MUSH method learns a more accurate representation of a paragraph with compared to the existing methods. In particular, it leverages and embeds sentiment information from sentence-level to paragraph-level by a hierarchical model. At sentence-level, MUSH adopts convolutional neural networks (CNN) with multi-size filters to reveal multi-scale sentiment atoms based on word embedding. Upon these sentiment atoms, a bidirectional gated recurrent neural network (RNN) is used to capture the inter-atoms interactions. At paragraph-level, to reveal the polarity-shift among sentences, a similar multi-scale CNN-RNN structure is utilized. This structure can recognize the multi-scale sentence-level sentiment segment by CNN and can exploit their dependence by RNN. As a result, the representation ability of MUSH is largely lifted by simultaneously disentangling the multi-forms and hierarchical polarity-shift compared with the existing methods. Meanwhile, by adopting the hierarchical model, MUSH also captures contextual, sequential, and sentiment information in a paragraph, which is essential for sentiment analysis, as analyzed in Section 2.2.

The key contributions made of this chapter include the following:

- *Learning polarity-shift with multiple forms:* The CNN with multi-size filters is learned to reveal the multi-scale sentiment atoms and sentiment segments. This learning model leverages multi-scale local information around a word or a sentence; thus, it comprehensively reveals the polarity-shift with different forms.

- *Learning hierarchical polarity-shift with complex dependence:* A hierarchical CNN-RNN structure is proposed to capture the polarity-shift at both sentence-level and paragraph-level. The local dependence is learned by multi-layer CNN, and the long-term dependence is captured by bidirectional gated RNN upon the CNN learned abstractions. The co-working of CNN and RNN in a hierarchical way effectively disentangles complex polarity-shifts and embeds them into a vector space.

- *An end-to-end paragraph-embedding model for polarity-shift-sensitive sentiment classification:* A model is proposed to learn an end-to-end paragraph embedding that not only considers contextual, sequential, and sentiment information but also captures polarity-shift, catering to complex real-world sentiment-analysis tasks.

This chapter compares MUSH with state-of-the-art paragraph-embedding methods on four large real-world data sets with various data characteristics. The experimental results demonstrate that MUSH significantly improves the sentiment-classification performance, especially for the paragraph with polarity-shift.

The rest of this chapter is organized as follows: Section 3.2 systematically analyzes the sentiment-polarity-shift characteristics. Section 3.3 gives the details of the proposed model MUSH. Section 3.4 illustrates the performance of MUSH by comparing it with state-of-the-art sentiment analysis methods in real-world data sets. Section 3.5 concludes this paper and discusses the further challenges and opportunities.

3.2 Sentiment-Polarity-shift Characteristics

This chapter mainly considers two primary characteristics of sentiment-polarity-shift that bring significant challenges for paragraph representation: (1) polarity-shift is with multiple forms, and (2) polarity-shift hierarchically exists in a paragraph. This section analyzes these two characteristics separately.

3.2.1 The Multiple Forms of Polarity-Shift

Polarity-shift has multiple forms that are caused by different factors, including *explicit indicators* and *implicit descriptions*. The explicit indicators are the words or terms that can cause a consistent or contrary polarity change. Typical explicit indicators include negative words and conjugation words. For example, the negative word ‘not’ is an explicit indicator. The positive sentiment polarity in ‘a good product’ will shift to the negative sentiment polarity if the explicit indicator ‘not’ is added (i.e., ‘not a good product’). In contrast to the explicit indicators, implicit descriptions do not involve any indicator in a paragraph. They change the sentiment polarity by a sequential combination of specific words or sentences. The following sentence gives an example of polarity-shift caused by implicit descriptions. ‘Yes, she is so beautiful that even the pockmarks on her face are shining brightly.’ The sentiment polarity in this sentence changes from positive to negative by the combination of ‘beautiful’ and ‘pockmarks are shining brightly.’

Although current efforts have tried to capture the explicit indicators, most of them fail to capture the explicit indicators and implicit descriptions simultaneously. As a result, they may obtain an incorrect sentiment of a paragraph when the paragraph has implicit descriptions that cause polarity-shift. This chapter proposes MUSH to solve this problem. The MUSH method jointly captures explicit indicators and implicit descriptions by a CNN-RNN structure. Accordingly, it can recognize multi-form polarity-shifts for sentiment analysis.

3.2.2 The Hierarchy of Polarity-Shift with Long-Term Dependence

Polarity-shift hierarchically exits in a paragraph from sentence-level (e.g., a negative word changes the sentiment polarity of its followed word, as discussed in Section 3.2.1) to paragraph-level (e.g., a transition sentence may change the sentiment polarity of its above sentence; 'Overall, I think this might be a good start for German food, but I prob wouldnt say its the best. However, if you're in the area and wanted to give German food a try, this place is decent.'). This hierarchy brings two significant challenges for sentiment analysis. The first challenge is that the sentiment polarity of a paragraph is hard to be recognized. This hardness exits because the sentiment polarity is determined by the interactions of polarity-shifts within and between sentences. The second challenge is that the long-term polarity-shift dependence of words or sentences is hard to be captured. For example, a word at the beginning of a sentence may shift the polarity of another word at the end of the sentence; however, it is hard to determine which words have this dependence. The above two challenges caused by hierarchical polarity-shifts prevent a robust sentiment analysis.

This chapter proposes MUSH to tackle these challenges. The MUSH method disentangles intra-level and inter-level polarity-shifts and their interactions. Consequently, it provides an in-depth understanding of a paragraph for robust sentiment analysis.

3.3 Multi-Scale and Hierarchical Network for Polarity-Shift Sentiment Analysis

This chapter proposes a multi-scale and hierarchical embedding method, MUSH, for polarity-shift sentiment analysis. The structure of MUSH is shown in Figure 3.1. The primary structure of MUSH is a hierarchical neural network that consists of sentence-level and paragraph-level embedding modules, each of which is a multi-scale CNN-RNN structure. The sentence-level embedding module captures the contextual information, sequential information, and multi-form sentence-level polarity-shifts. The paragraph-level embedding module reveals the sentiment segments and multi-form paragraph-level polarity-shifts. Accordingly, MUSH preserves the distribution properties of a paragraph with explicit polarity-shift modeling. The MUSH method learns a paragraph representation through an end-to-end training process guided by a specific sentiment classification task to further capture the sentiment information. It then feeds the learned paragraph representation into the well-trained sentiment classifier for robust sentiment analysis.

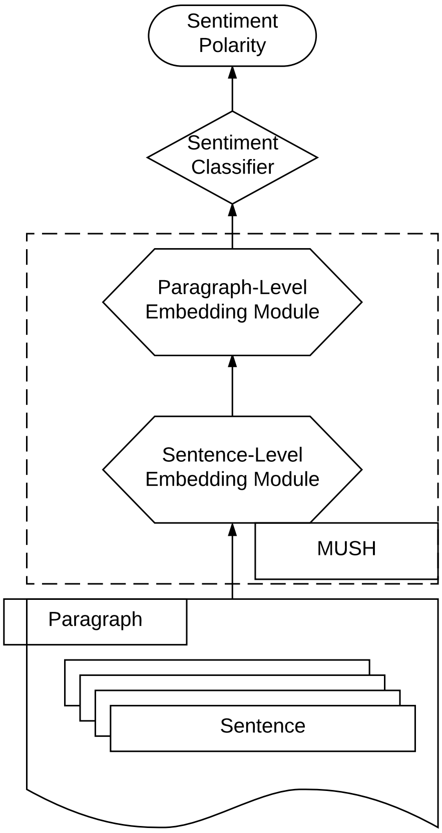

Sentiment Polarity
Sentiment Classifier
Paragraph-Level Embedding Module
Sentence-Level Embedding Module
MUSH
Paragraph
Sentence

Figure 3.1: The MUSH Structure: The primary structure of MUSH is a multi-scale and hierarchical neural network that consists of two modules (i.e., sentence-level embedding module and paragraph-level embedding module). The MUSH method learns paragraph representation for polarity-shift-sensitive classification through an end-to-end training process guided by a specific task.

3.3.1 Sentence-Level Embedding Module

The sentience-level embedding module learns the word embedding function e_w and sentence embedding function e_s. The structure of this module is shown in Figure 3.2. The first layer of this module is a word embedding layer. This word embedding layer should be pre-trained by an unsupervised method that captures word contextual information, such as skip-gram Mikolov et al. (2013), to generate context-based word representation. Although the context-based word representation may not provide meaningful information for sentiment polarity discrimination, it reflects the local structures in a paragraph that contributes to the generalization ability of the final paragraph representation. After the first layer, MUSH adopts CNN layers with multi-size filters to capture multi-scale sentiment atoms. The CNN layers reveal multi-scale local patterns in a sentence. These multi-scale local patterns are similar to the n-gram features used by traditional NLP methods. However, they have higher utility to the final task compared to n-gram features because these patterns are driven by a sentiment analysis task while n-gram features are not all related to sentiment analysis. After each CNN layer, MUSH uses a bidirectional gated RNN layer to leverage the dependence of the CNN learned sentiment atoms. Accordingly, different bidirectional gated RNNs capture polarity-shift in different scales. Finally, MUSH merges the outputs of RNN layers by fully connected layers, which reveal the relationships among multi-scale polarities.

3.3.2 Paragraph-Level Embedding Module

The paragraph-level embedding module learns the paragraph embedding function e_d. The structure of this module is shown in Fig 3.3. In this module, MUSH adopts a CNN-RNN network that has the same structure of the CNN-RNN network used in the sentence-level embedding module. However, these CNN-RNN networks capture different information. Specifically, in the paragraph-level embedding module, the multi-scale CNN layers capture sentiment segments in a paragraph, which reflect the multi-scale local sentence polarity dependencies. Then, MUSH adopts a bidirectional gated RNN to capture the polarity-shift in the CNN captured sentiment segments. The rationale for the adoption of the CNN-RNN network is as follows. Since CNN suits for discovering local patterns with fixed receive filed, MUSH adopts CNN with multi-size filters to capture information in different receive fields that complement each other. Considering RNN can capture sequential information but does not suit for capturing long-term dependencies, MUSH decomposes a paragraph into hierarchical parts to cut down the length of a long sequence. By this means, MUSH effectively presents the sentiment polarity of a paragraph by utilizing the language structure of a paragraph and the natural properties of polarity-shifts.

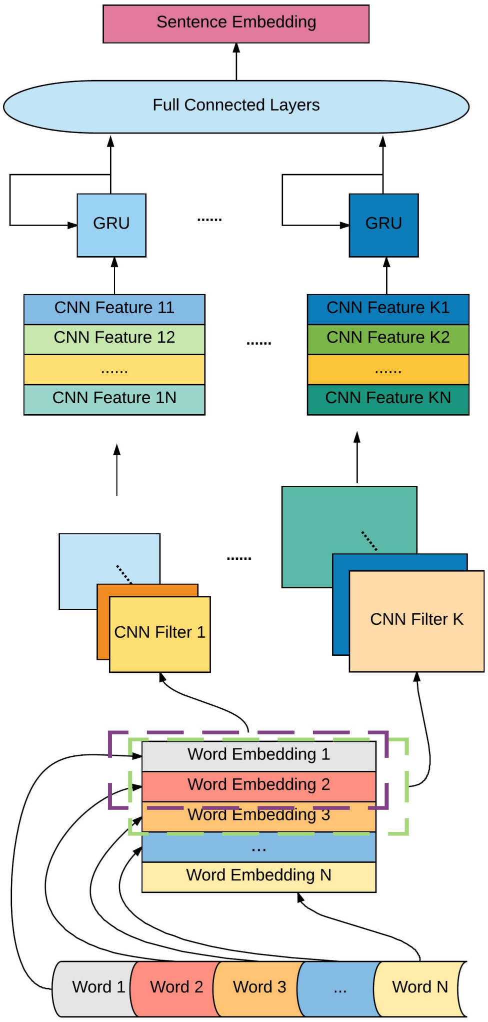


Figure 3.2: The Sentence-level Embedding Module Structure: The first layer of this module is a word embedding layer. This word embedding layer then connects to several CNN layers with multiple filter sizes. Each of these CNN layers follows a bidirectional gated RNN layer. The outputs of these bidirectional gated RNN layers are merged by fully connected layers to form a sentence representation.

3.4 Experiments and Evaluation of Multi-Scale and Hierarchical Network

The experiments evaluate the sentiment analysis performance of MUSH comparing with five state-of-the-art methods and eight baseline methods on four large real-world data sets.

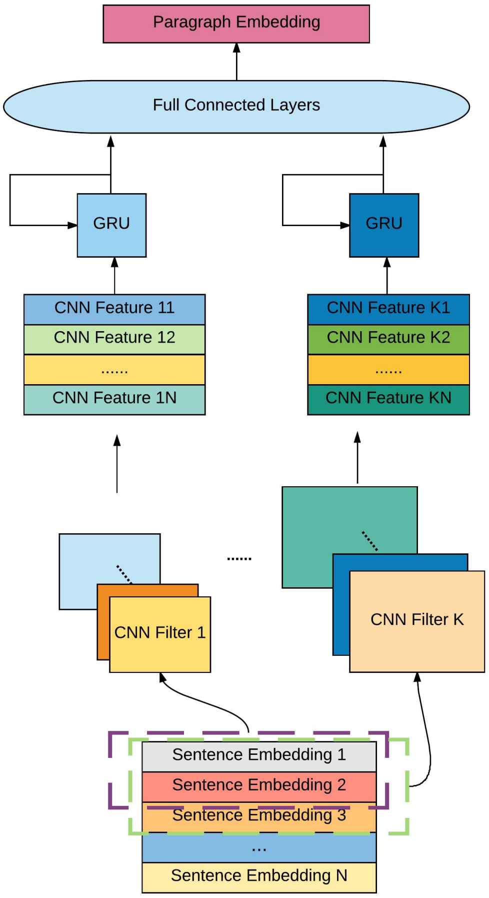


Figure 3.3: The Paragraph-level Embedding Module Structure: The CNN layers with multiple filter sizes followed by bidirectional gated RNN layers are used to capture multi-scale polarity-shift. The fully connected layers merge the outputs of RNN layers to form a paragraph representation.

3.4.1 Data sets

The data sets used in the experiments include *IMDB movie review*, *Yelp 2013*, *Yelp 2014*, and *Yelp 2015*. These data sets are widely used as benchmarks to evaluate sentiment classification performance.

The first data set, *IMDB movie review*, was collected by Diao et al. (2014). It involves 348,415 movie reviews with sentiment rating levels ranging from 1 to 10. The *Yelp 2013*, *Yelp 2014*, and *Yelp 2015* data sets are provided by Yelp challenge in 2013, 2014, 2015, respectively. They contain public hotels and restaurants comments with 335,018, 1,125,457, and 1,569,264 reviews in each year with rating levels ranging from 1 to 5. The details of these four data sets are shown in Table 3.1.

Table 3.1: Characteristics of Data Sets in the Experiments: #v refers to the number of vocabulary, #s and #w refers to the number of sentences and words, respectively.

data set	#class	#reviews	average #s	max #s	average #w	max #w	#v
IMDB	10	348,415	14.0	1,484	325.6	2,802	115,831
Yelp 2013	5	335,018	8.9	151	151.6	1,184	211,245
Yelp 2014	5	1,125,457	9.2	151	156.9	1,199	476,191
Yelp 2015	5	1,569,264	9.0	151	151.9	1,199	612,636

3.4.2 Experimental Settings

3.4.2.1 Comparison Methods

The experiments compare MUSH with four baseline methods and three state-of-the-art methods to evaluate MUSH performance. The baseline methods include bog-of-words (BOW), bag-of-words with term frequency-inverse document frequency (BOW-TFIDF), ngram, and ngram with term frequency-inverse document frequency (ngram-TFIDF). The features generated by these methods are fed into a logistic regression to form sentiment classifiers. The state-of-the-art embedding methods and their variants include:

- *SentCNN* (Kim 2014). This method captures neighbourhood relationships among words by feeding word embedding into CNN. According to the trainability of the word embedding, *sentCNN* derives four variants: *sentCNN-random*, *sentCNN-static*, *sentCNN-nonstatic* and *sentCNN-multi*. *SentCNN-random* uses a random vector to embed each word. *SentCNN-static*, *sentCNN-nonstatic* and *sentCNN-multi* uses the skip-gram method (Mikolov et al. 2013) for word embedding. While *sentCNN-static* fixes the word embedding, *sent-nonStatic* dynamically adjusts the word embedding when training CNN. In contrast, *sentCNN-multi* uses both static and dynamic word embedding to learn text representation.

- *GateRNN* (Tang, Qin & Liu 2015). GateRNN learns inter-sentence relationships based on sentence embedding via the recurrent neural network with gated recurrent units. Different sentence embedding models, i.e. CNN and LSTM, induce two *gateRNN* variants: *gateRNN-CNN* and *gateRNN-LSTM*.

- *HNATT* (Yang et al. 2016). *HNATT* integrates word and sentence-level sequential characteristics through a hierarchical network. Different integrating methods induce three *HNATT* variants: *HNATT-ATT*, *HNATT-AVG* and *HNATT-MAX*. *HNATT-ATT* uses attention mechanism to weighted sum words and sentences embedding. *HNATT-AVG* and *HNATT-MAX* adopt the averaged and max value of words/sentences embedding, respectively.

3.4.2.2 Data Preprocessing

The pre-processing method for a paragraph is following the setting in the work of Yang et al. (2016). Stanford's CoreNLP (Manning et al. 2014) is used to split a paragraph into sentences and tokenize each sentence. The word embedding is initially set according to the pre-training via the skip-gram model (Mikolov et al. 2013).

3.4.2.3 Neural-Network Implementation

In the experiments, the word embedding dimension is set as 100. Regarding MUSH, the CNN with 2×100 and 3×100 filter sizes are used to capture the 2-gram and 3-gram features. In the sentence-level embedding module, the number of filters in CNN layers, the number of gated recurrent units (GRU), and the number of nodes in the fully-connected layer is set as 32, 64, and 64, respectively. In the paragraph-level embedding module, these settings are 64, 128 and 128, respectively. A single layer feed-forward neural network (SLFN) with softmax is used as the sentiment classifier. In the training process, the batch normalization is used after every layer, and the dropout with keeping probability 0.5 is adopted after fully connected layers. We set the batch size as 64 and use Adam algorithm (Kingma & Ba 2014) to optimize the MUSH.

3.4.3 Evaluation on Sentiment Classification Performance Enabled by Multi-Scale and Hierarchical Network

This experiment compares the sentiment classification performance enabled by MUSH with that enabled by baseline and state-of-the-art methods and their variants that are discussed in Section 3.4.2. The sentiment classification results on different data sets are shown in Table 3.2 with respect to accuracy (%). In Table 3.2, the best result for each data set is shown in boldface.

As can be seen in Table 3.2, MUSH achieves the best results in all four data sets. The performance lift is mainly benefited by (1) hierarchical modeling the paragraph, and (2) capturing polarity-shift. From the results, MUSH is significantly better than all competitors instead of HNATT. It is because HNATT is also a hierarchical model that shares the same strengths as MUSH. Besides, HNATT uses the attention mechanism to capture critical information in a paragraph. Accordingly, HNATT can also focus

Table 3.2: Sentiment Classification Accuracy of Different Methods

Methods	IMDB	Yelp 2013	Yelp 2014	Yelp 2015
BOW	15.5	20.2	29.4	31.8
BOW-TFIDF	21.3	31.6	36.0	38.9
ngram	25.1	36.6	41.3	44.5
ngram-TFIDF	27.3	39.1	43.2	45.7
sentCNN-random	29.4	55.6	57.9	58.9
sentCNN-static	29.2	53.5	58.2	59.2
sentCNN-nonstatic	24.6	47.1	52.6	52.2
sentCNN-multi	29.4	53.3	58.3	58.7
gateRNN-CNN	40.6	61.4	62.7	63.2
gateRNN-LSTM	41.7	63.5	66.9	67.8
HNATT-ATT	42.5	63.2	67.3	68.1
HNATT-AVG	42.2	63.8	66.8	67.7
HNATT-MAX	42.9	63.7	66.8	68.0
MUSH	**43.2**	**64.3**	**67.5**	**68.9**

on polarity-shift indicators to some extent. However, MUSH adopts a direct way to model and capture polarity-shift. It provides a better understanding of polarity-shift complexities. Therefore, MUSH gets a better embedding performance than HNATT.

3.4.4 Evaluation on Polarity-Shift Capturing of Multi-Scale and Hierarchical Network

This experiment evaluates the polarity-shift-capturing ability of MUSH and whether this polarity-shift-capturing ability contributes to sentiment classification. It demonstrates the insight into the superior sentiment classification performance of MUSH. To achieve this goal, this experiment evaluates the polarity-shift-capturing ability of MUSH from two aspects: (1) embedding the polarity-shift triggered by explicit indicators; and (2) embedding the polarity-shift triggered by implicit descriptions. For each aspect, the experiment first constructs texts that have the corresponding polarity-shift and then analyzes the sentiment of these texts by MUSH and its competitors[1] that trained on the

[1] This experiment reports only the result of one variant for each sentiment analysis method. Although variants of a method may have a slight difference, they do not produce significantly different results in this testing because they share the same foundation and design.

Yelp 2015 data set.

Table 3.3 shows the constructed texts and the sentiment analysis results. On the Yelp 2015 data set, the ground-truth sentiment polarity is manually assigned by users and ranges from 1 to 5, where 1 indicates the strongest negative sentiment and 5 indicates the strongest positive sentiment. Accordingly, the sentiment polarity predicted by each sentiment analysis method in this experiment also ranges from 1 to 5. A larger sentiment rating indicates a stronger positive sentiment, and vice versa.

Table 3.3: Polarity-Shift Capturing Effectiveness of Different Methods. The predicted sentiment ratings are reported. A larger rating indicates a stronger positive sentiment, and vice versa.

Text	ngram_TFIDF	SentCNN_static	GateRNN_LSTM	HNATT	MUSH
I feel this restaurant is good	4	4	4	5	3
I feel this restaurant is *not* good	1	4	1	1	1
I *don't* feel this restaurant is good	4	4	4	5	1
I feel this restaurant is *so* good	5	4	5	5	5
Not sure ⋯ the bread and the rice[2].	2	4	2	1	1

The experimental results indicate that MUSH embeds explicit indicators better than its competitors. For demonstration, Table 3.3 constructs a text, "*I think this hotel is good.*", and its counterparts with the polarity-shift triggered by explicit indicators, including adverb and disjunctive. The sentiment of "*I think this hotel is good.*" is treated as positive by all methods except MUSH. Actually, the sentiment polarity of this sentence is not strongly positive. It can be further emphasized by adding some adverb word, for example "*so*", "*I think this restaurant is so good.*" before the adjective "*good*". Indeed, when the explicit polarity-shift indicator "*so*" is added, MUSH certainly assigns positive sentiment polarity to this sentence, which reflects MUSH is sensitive to this kind of polarity-shift. This advantage is contributed by the multi-scale CNN to reveal the multi-forms polarity-shift. MUSH can also well capture the polarity-shift caused by disjunctive and adjective pairs. When disjunctive "*not*" is added, all methods except *sentCNN* assign negative sentiment to sentence "*I think this hotel is not good*". However, only MUSH treats sentence "*I don't think this hotel is good.*" as negative. This demonstrates that most competitors only capture the polarity between a nearby disjunctive-adjective pair, e.g. "not" and "good". However, MUSH can capture the polarity-shift between a distant disjunctive-adjective pair, which is a kind of long-term dependence captured by the RNN layer in MUSH.

[2] Not sure why this was on food network. Chicken was fatty and dark meat in curries. The best parts of the meal were the bread and the rice.

The experimental results indicate that MUSH and *HNATT* can effectively capture the polarity-shift triggered by implicit descriptions. To show that, we pick up a text, "*Not sure why this was on food network. Chicken was fatty and dark meat in curries. The best parts of the meal were the bread and the rice.*", from the Yelp 2013 data set. The text contains a negative sentiment that hidden in the interactions of sentences. The sentiment rating of the text is 1. For this text, MUSH and *HNATT* accurately predict its sentiment rating as 1, while other methods fail to predict the right sentiment rating. Specifically, *ngram-TFIDF*, *sentCNN*, and *gateRNN* predict it as 2, 4, and 2, respectively. These results are driven by the fundamental prior to these embedding methods. While *ngram-TFIDF*, *sentCNN*, and *gateRNN* overlook the relationships between sentence, MUSH and *HNATT* consider this kind of implicit polarity-shift descriptions. Different from *HNATT*, which only captures the global interactions by RNN, MUSH further models the local multi-forms polarity-shift by multi-scale CNN that enables better performance.

3.5 Summary

This chapter proposes a multi-scale and hierarchical embedding method, MUSH, for polarity-shift sentiment analysis. The MUSH method uses an end-to-end learning approach to embed a paragraph into a vector space that recognizes the polarity-shift and maintains the context information, sequential information, and sentiment information in paragraph space for sentiment classification. The experimental results support the performance merits of the MUSH compared with state-of-the-art methods. The MUSH method captures the sentiment features in an implicit process. In the next chapter, the explicit sentiment information and features will be considered to combine with the implicit MUSH features to construct more powerful sentiment representation.

CHAPTER 4

Sentiment Analysis on Non-Independent and Identical Distributed Paragraph

4.1 Introduction

In real-life textual data, words and sentences are coupled with each other (non-independent), and they may have heterogeneous meanings under different contexts or in different language locales (non-identically distributed). In other words, real-life textual data has non-IID characteristics. As analyzed in Section 2.2, non-IID characteristics essentially determine the sentiment of a text yet are hard to be captured. This chapter focuses on the sentiment analysis on a paragraph with such non-IID characteristics (a.k.a., *non-IID paragraph*), considering a non-IID paragraph contains non-IID sentences. This chapter contributes to the reliable sentiment analysis in terms of capturing complex interactions in a social media paragraph but does not consider the sentiment-polarity-shift, which has been studied in Section 3.

Learning non-IID characteristics has attracted increasing attention and enabled significant performance gain in a variety of applications and learning tasks, such as classification (Zhu et al. 2018), clustering (Jian et al. 2018), outlier detection (Xu et al. 2018, Meng et al. 2019), and image processing (Shi et al. 2014). However, only limited effort has been made in non-IID paragraph representation. For example, Teng et al. (2016) aim to capture the nearby word couplings, Tang, Qin & Liu (2015) focus on the sequential relations of words, and Wang, Huang, Zhu & Zhao (2016) model the hierarchical heterogeneous meanings of words and sentences. Although the existing non-IID paragraph representation methods significantly improve the performance of sentiment analysis, they do not comprehensively capture the non-IID characteristics in an effective way. As a result, they may fail to accurately represent a non-IID paragraph, even if the paragraph has a clear sentiment polarity. For example, most of the current sentiment analysis on non-IID paragraph may classify the sentiment polarity of both "I feel the restaurant is *good*" and "I *don't* feel the restaurant is *good*" as positive because these methods may only focus on the sentiment word "good" but ignore the dependence (non-independent) between "don't" and "good". Also, they treat the polarity of "The taste is *great*" and "Cost a *great* deal of time" as equal since they may overlook the different meanings (non-identically distributed) of "great" in different contexts. Comprehensively capturing the non-IID characteristics requires a complex model, in which, however, the high model complexity may easily cause over-fitting (Spiegelhalter et al. 2002).

This chapter studies the nature of paragraph non-IID characteristics and proposes a novel non-IID paragraph-representation framework to comprehensively model the non-IID characteristics for sentiment analysis. Specifically, the proposed framework systematically captures the hierarchical non-IID characteristics (i.e., the coupling relations and heterogeneous meanings of words and sentences) and hierarchically embeds them into vector representations, termed as implicit features. It further regulates the implicit features by explicit features, which are designed by sentiment priors, to avoid over-fitting and capture additional sentiment-related information. This framework is instantiated by a **m**ulti-scale and hi**e**rarchical **de**ep neural network with an**a**ttention mechanism (MEDEA). Specifically, MEDEA exploits hierarchical and heterogeneous coupling relations by adopting a multi-scale CNN-RNN structure and captures the heterogeneous meanings of words and sentences with an attention mechanism. Besides, it hierarchically integrates implicit features and explicit features from the word-level to the paragraph-level according to semantic matching.

The key contribution made in this chapter include the following:

- *A non-IID paragraph-representation framework for sentiment analysis*, which models both the hierarchical coupling relations and the heterogeneous meanings of words and sentences. These coupling relations and heterogeneous meanings essentially determine the sentiment polarity of a non-IID paragraph. As far as we know, the proposed framework is the first non-IID paragraph-representation framework that comprehensively models the non-IID characteristics for sentiment analysis.

- *A novel deep neural network structure for modeling non-IID paragraphs*, by a hierarchical multi-scale CNN-RNN module with an attention mechanism. The multi-scale CNN-RNN structure learns the hierarchical coupling relations from the word-level to the paragraph-level to disclose the complex sentiment in a paragraph. The attention mechanism learns the heterogeneous meanings of words and sentences to disentangle sentiment ambiguities in text and induce a more precise paragraph representation.

- *Hierarchically integrating explicit features and implicit features to avoid over-fitting*. While the implicit features reveal the sentiment polarity of a non-IID paragraph, the explicit features reduce the risk of over-fitting in the implicit features' learning process and provide additional sentiment information instead of that learned from sentiment labels.

This chapter conducts comprehensive experiments on five large real-world data sets, including IMDB Diao et al. (2014), Amazon Jindal & Liu (2008), and three data sets from *Yelp Challenges* (i.e., Yelp13, Yelp14, and Yelp15) to show the complex non-IID paragraph characteristics and evaluate the performance of the proposed method. It also evaluates the proposed method on two small real-world data sets from Twitter sentiment analysis tasks (i.e., Twitter and Twitter-air). The experimental results show that (1) the proposed MEDEA network learns non-IID paragraph characteristics effectively, (2) MEDEA enjoys performance gain (up to 1.76% in terms of accuracy, 5.26% in terms of RMSE and 5.93% in terms of RMSE@K) from the learned non-IID characteristics, and (3) MEDEA can be trained easily and can achieve large accuracy improvement compared with its variants that only with implicit features (up to 7.86% in terms of accuracy) by hierarchically integrating explicit features and implicit features, resulting significant performance enhancement (up to 18.16% in terms of RMSE and 11.19% in terms of RMSE@K). These results strongly evidence that the proposed non-IID paragraph-representation framework suits for sentiment analysis, and the deep neural network instantiation of this framework achieves superior sentiment analysis performance.

In the rest of this chapter is organized as follows: Section 4.2 comprehensively analyzes the non-IID characteristics in a paragraph and proposes the non-IID paragraph-representation framework. Section 4.3 gives the details of the technical mechanism and components of MEDEA. Section 4.4 empirically evaluates MEDEA in different aspects. Section 4.5 concludes the paper and discusses promising prospects.

4.2 Non-Independent and Identically Distributed Paragraph Representation Design

This section first formalizes the learning problem and the learning objective of the non-IID paragraph representation for sentiment analysis. It then deeply analyzes the non-IID characteristics of a paragraph. Based on the analyzed non-IID characteristics, it further proposes a non-IID paragraph representation framework.

4.2.1 Characteristics of Non-Independent and Identically Distributed Paragraph

A non-IID paragraph has two primary characteristics: coupling and heterogeneity. In a non-IID paragraph, the coupling means that words or sentences have complex interactions, and the heterogeneity refers to the different meanings of words or sentences in different language locales. These two characteristics form the so-called non-IIDness (Cao 2013), which essentially determines its sentiment polarity.

For a non-IID paragraph, the non-IID characteristics may hierarchically exist from word-level to paragraph-level. These non-IID characteristics are demonstrated in Figure 4.1. At word-level, coupling relations exist between a word and its neighbors (i.e., several words located around a word). For example, “not" and “sure" are coupled. These coupling relations constitute the basic sentiment elements in a sentence. Such sentiment elements are further coupled with each other directly (e.g., “not sure" and “why this") or indirectly (e.g., “best parts" and “bread and the rice") to determine the sentiment polarity of a sentence. Furthermore, a word may hold heterogeneous meanings in different language contexts and locales. For example, “dark" has negative sentiment polarity in the example, but it may have a neutral sentiment polarity if it is used as a color. At sentence-level, similar non-IID characteristics as that at word-level also exist. A sentence interacts with its neighbors (e.g., in the example, the first and second sentences are coupled), and these interactions further couple with each other (e.g., the indirect couplings indicated in the example) to determine the sentiment polarity of a paragraph. Besides, a sentence may have different sentiment polarities in different locales. For example, the last sentence in the example has a negative sentiment polarity considering the first and second sentences, but it may have a positive sentiment polarity if only this sentence itself is evaluated.

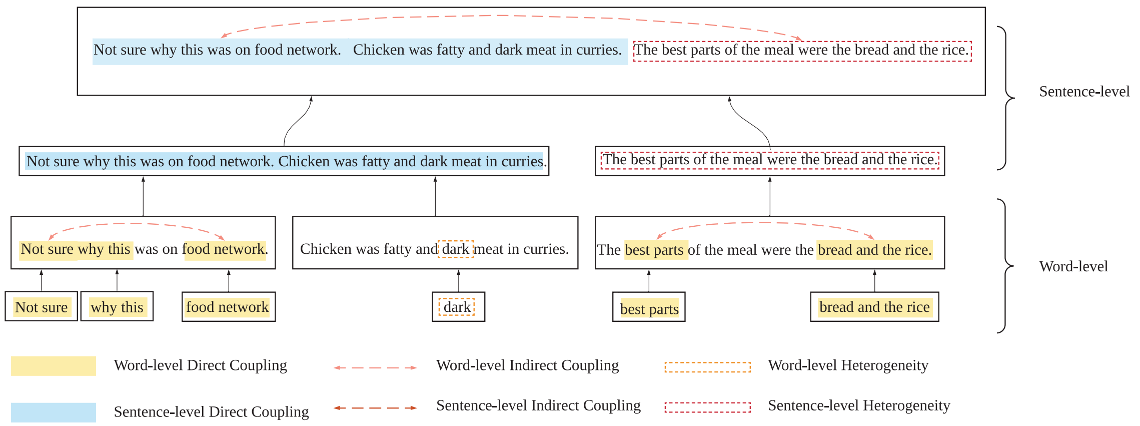

Not sure why this was on food network. Chicken was fatty and dark meat in curries. The best parts of the meal were the bread and the rice.
Sentence-level
Not sure why this was on food network. Chicken was fatty and dark meat in curries.
The best parts of the meal were the bread and the rice.
Not sure why this was on food network.
Chicken was fatty and dark meat in curries.
The best parts of the meal were the bread and the rice.
Word-level
Not sure
why this
food network
dark
best parts
bread and the rice
Word-level Direct Coupling
Word-level Indirect Coupling
Word-level Heterogeneity
Sentence-level Direct Coupling
Sentence-level Indirect Coupling
Sentence-level Heterogeneity

Figure 4.1: An Example of the Non-IID Characteristics in a Paragraph.

4.2.2 Objectives of Non-Independent and Identically Distributed Paragraph Representation

Given a paragraph $P \in \mathbb{P}$ that consists of a sequence of n_s sentences $\{s_i | i = 1, \cdots, n_s\}$, the i-th of which consists of a sequence of n_{wi} words $\{w_{i,j} | j = 1, \cdots, n_{wi}\}$, a paragraph representation model is a function $E(\cdot) : \mathbb{P} \to \mathbb{R}^{n_f}$ that transforms the paragraph P to a vector $\mathbf{p} \in \mathbb{R}^{n_f}$ with n_f dimensions. Here, $\mathbb{P}$ refers to a paragraph space, and $\mathbb{R}$ refers to a real space.

Denoting the sentiment polarity of the j-th word in the i-th sentence of a paragraph P as $o_{i,j}$, the polarity of the i-th sentence is denoted as $O_i = \oint_1^{n_{wi}} o_{i,j} dw_{i,j}$. Here, $\oint_1^{n_{wi}} dw_{i,j}$ refers to a sequential operation from $w_{i,1}$ to $w_{i,n_{wi}}$. Consequently, the polarity $O \in \mathbb{O}$ of a paragraph P is denoted as $O = \hat{\oint}_1^{n_s} O_i ds_i$. The $\hat{\oint}_1^{n_s} ds_i$ is a sequential operation from s_1 to s_{n_s}, and this operation is different from the one in $\oint_1^{n_{wi}} dw_{i,j}$. The polarity of a paragraph is determined by the operation $\oint$ and $\hat{\oint}$ from the word level to the sentence level, that is $\hat{\oint}\oint : \mathbb{O}^{n_s \times n_{wi}} \to \mathbb{O}$. Here, $\mathbb{O}$ refers the sentiment polarity space.

When considering paragraph representation for sentiment analysis, the objective is to learn a representation function $E(\cdot)$ that can effectively reflect the sentiment polarity of a paragraph per a sentiment classifier $C(\cdot) : \mathbb{R}^{n_f} \to \mathbb{O}$. Formally, denoting the distribution of the polarity of a set of paragraph $\mathbf{P}$ as $\mathfrak{D}$, the learning objective function of paragraph representation for sentiment analysis can be written as follows:

$$\underset{E(\cdot),C(\cdot)}{\text{minimize}} \quad Div(\mathfrak{D} || \{C(E(P)) | P \in \mathbf{P}\}), \tag{4.1}$$

where $Div(\cdot||\cdot)$ is a divergence for measuring the difference between two distributions.

To fit different data characteristics in different tasks, $Div(\cdot||\cdot)$ can be instantiated by different divergence measures or transformed divergence functions, such as KL-divergence, cross-entropy, and Hellinger distance. The intuition behind this learning objective function is that the sentiment statistics of paragraphs should be preserved in their representation space. In other words, the sentiment statistics should be recovered by a classifier $C(\cdot)$ from the representation space.

In practice, the objective function Equation (4.1) may be accompanied by one of the following two problems: *high generalization error* and *low model fitness*. The generalization error refers to the gaps between the sentiment distribution in paragraphs and that learned by the classifier. High generalization error is mainly caused by lacking training samples. Although thousands of millions of paragraphs are available for training, the number of training samples is still not sufficient to reflect the whole paragraph statistics. The model fitness refers to the ability of representation function $E(\cdot)$ to represent the sentiment-related information in paragraphs. Low model fitness is often caused by ignoring complex sentiment-related data characteristics of paragraphs when designing the representation function $E(\cdot)$.

In order to reduce the generalization error, a promising way is to preserve some distribution properties of a paragraph space $\mathbb{P}$ in its representation space $\mathbb{E}:\mathbb{R}^{n_f}$. With this constraint, denoting the distribution of a paragraph in space $\mathbb{P}$ as $\mathfrak{P}$ and in space $\mathbb{E}$ as space $\mathfrak{E}$, the learning objective function can be revised as follows:

$$\underset{E(\cdot),C(\cdot)}{\text{minimize}} \quad Div(\mathfrak{D}||\{C(E(P))|P \in \mathbf{P}\}) + \gamma \hat{Div}(\mathfrak{P}||\mathfrak{E}), \tag{4.2}$$

where $\hat{Div}$ is a divergence measure for two distributions, and γ is a trade-off parameter. This distribution preservation can be achieved by introducing additional sentiment information, such as explicit features designed by the prior of sentiment analysis.

In order to enhance the model fitness, one approach is to model the data characteristics that determine the sentiment polarity. This chapter hypothesizes that interactions between words and sentences essentially determine the sentiment polarity of a paragraph. These interactions are involved in the operations $\oint$ and $\hat{\oint}$. Here, we denote the word-representation function as $E_w(\cdot):\mathbb{W}\to\mathbb{R}^{n_{ew}}$, sentence-representation function as $E_s(\cdot):\mathbb{R}^{n_{wi}\times n_{ew}}\to\mathbb{R}^{n_{es}}$, and paragraph-representation function as $E_p(\cdot):\mathbb{R}^{n_s\times n_{es}}\to\mathbb{R}^{n_f}$, where $\mathbb{W}$ is the word space, the n_{ew} is the dimension of the word-representation space, and the n_{es} is the dimension of the sentence representation space. To model such interactions, the representation function $E(\cdot)$ should be decomposed by representation functions from the word level to the paragraph level with sequential operations. In this way, the learning objective function should be reformulated as follows:

$$\underset{E_w(\cdot),E_s(\cdot),E_p(\cdot),C(\cdot)}{\text{minimize}} \quad Div(\mathfrak{O}||\{C(E_p(E_s(E_w(w))))|w \in s, s \in P, P \in \mathbf{P}\}). \tag{4.3}$$

This chapter jointly considers the distribution preservation and the non-IID characteristics for a more accurate paragraph representation for sentiment classification. Combining Eqs. (4.2) and (4.1), the learning objective of this model is formalized as:

$$\underset{E_w(\cdot),E_s(\cdot),E_p(\cdot),C(\cdot)}{\text{minimize}} \quad Div(\mathfrak{O}||\{C(E_p(E_s(E_w(w))))|w \in s, s \in P, P \in \mathbf{P}\}) + \gamma \hat{Div}(\mathfrak{P}||\mathfrak{E}). \tag{4.4}$$

4.2.3 Architecture for Representing Non-Independent and Identically Distributed Paragraph

Following the analysis of the non-IID characteristics and the learning objective function Equation (4.4), this chapter proposes a non-IID paragraph representation framework for sentiment analysis. It illustrates this framework in Figure 4.2. The non-IID paragraph representation framework has a hierarchical structure. It represents a paragraph from the word-level to the paragraph-level by the word-representation function $E_w(\cdot)$, the sentence-representation function $E_s(\cdot)$, and the paragraph-representation function $E_p(\cdot)$ in the objective function Equation (4.4).

Specifically, the word-representation function $E_w(\cdot)$ is implemented by an implicit-word-embedding layer, an explicit-word-feature layer, and fully-connected layers. The implicit-word-embedding layer captures word contextual information and embeds this information into a vector space as implicit word features. The explicit-word-feature layer extracts word features by designated word feature functions, which contain sentiment priors. Further, the fully-connected layers integrate these implicit and explicit word features to form a word-representation vector. The sentence-representation function $E_s(\cdot)$ is implemented by a non-IID-characteristics-learning module. This non-IID-characteristics-learning module captures the coupling relations and heterogeneous meanings of words and embeds them to a sentence-representation vector, the structure of which will be detailed in Chapter 4.3. Finally, the paragraph-representation function $E_p(\cdot)$ is implemented by a non-IID-characteristics-learning module, an explicit-paragraph-feature layer, and fully-connected layers. Here, the non-IID-characteristics-learning module has the same structure as that in the sentence-representation function; however, it generates implicit features of a paragraph by capturing the coupling relations and heterogeneous meanings of sentences instead of words. The explicit-paragraph-feature layer represents a paragraph by specific paragraph-feature functions, which are designed by sentiment

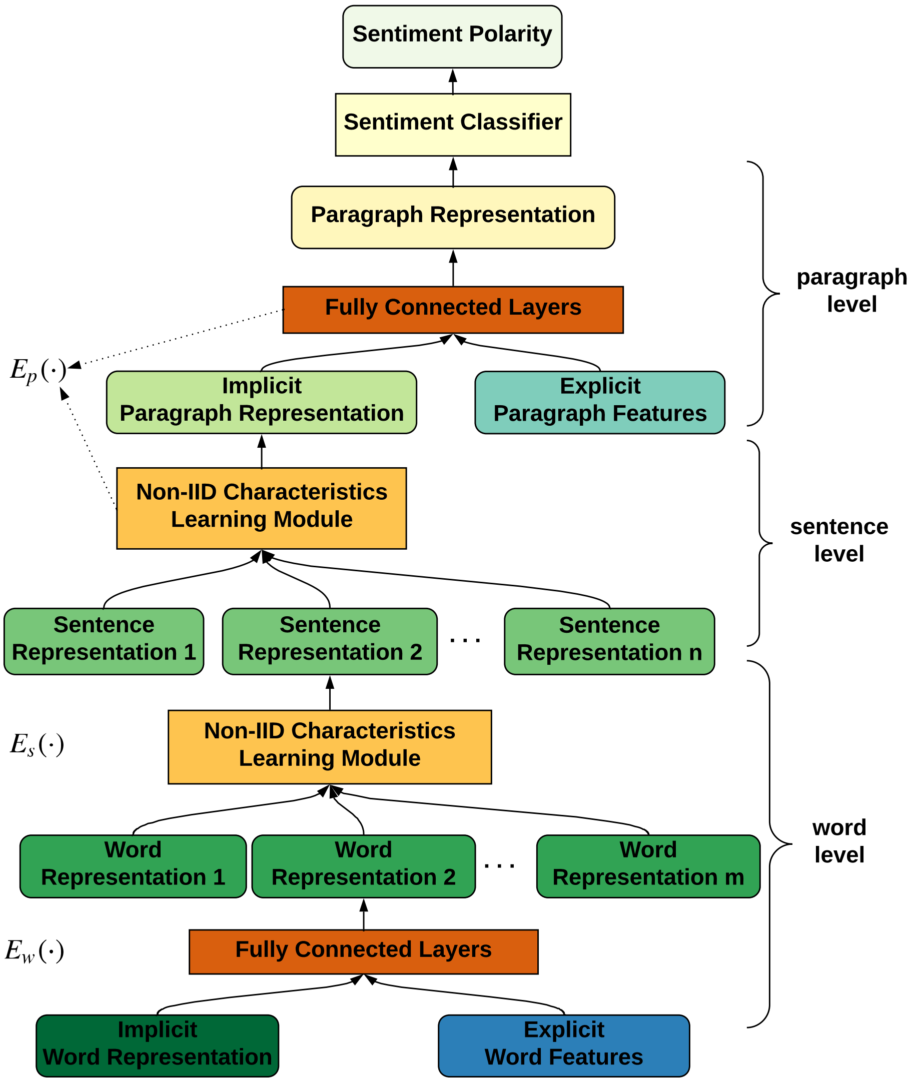


Figure 4.2: The Non-IID Paragraph Representation Framework for Sentiment Analysis.

priors. In the end, the fully-connected layers combine implicit and explicit paragraph features to constitute a non-IID paragraph-representation vector.

The non-IID paragraph representation framework implements the first term of Equation (4.4) by minimizing the divergence between the sentiment distribution of a paragraph and that predicted by the framework, and it implements the second term of Equation (4.4) through hierarchically involving prior-driven explicit features to preserve the paragraph distribution. In this way, the non-IID paragraph-representation framework not only captures the complex non-IID characteristics in a paragraph but also avoids over-fitting.

The workflow of the non-IID paragraph representation framework is as follows. For a paragraph, the non-IID paragraph representation framework first transforms each word in a paragraph to a word-representation vector through a pre-trained word-embedding method (e.g., skip-gram and CBOW; Mikolov et al. 2013). It then integrates each word-representation vector with prior-driven explicit sentiment features of the corresponding word by fully-connected layers. After that, it embeds non-IID characteristics of words and sentences into an implicit-paragraph-representation vector by two non-IID-characteristics-learning modules, respectively. Finally, the framework combines the implicit-paragraph-representation vector with pre-designed explicit paragraph features through fully-connected layers to construct a non-IID paragraph-representation vector. The framework learns non-IID paragraph representation through an end-to-end training process. Specifically, for a given sentiment analysis task, a sentiment classifier is built on the non-IID paragraph representation. Then, the sentiment labels of this task will guide the training of the non-IID paragraph representation framework. In other words, the learned non-IID paragraph representation will fit this sentiment analysis task well. Consequently, this non-IID paragraph representation framework incorporates the non-IID characteristics (i.e., hierarchical coupling relations and heterogeneous meanings of words and sentences) with explicit sentiment priors. It can be easily instantiated by designing suitable network structures for non-IID-characteristics-learning modules and selecting appropriate prior-driven features at both the word and paragraph levels.

4.3 Implement of Non-Independent and Identically Distributed Paragraph Representation Design

This chapter instantiates the non-IID paragraph representation framework as a multi-scale and hierarchical deep neural network with an attention mechanism, namely

MEDEA. Specifically, MEDEA implements the non-IID-characteristics-learning module as an attentive multi-scale CNN-RNN network and introduces three types of sentiment-related textual features as the prior-driven features.

4.3.1 Attentive Multi-Scale Convolutional-Recurrent Neural Network

The architecture of the attentive multi-scale CNN-RNN network is illustrated in Figure 4.3. The input of the attentive multi-scale CNN-RNN network is a set of representation vectors of words in a sentence or a set of representations vectors of sentences in a paragraph. The network first adopts an attention mechanism to transform the inputted representation vectors to attentive representation vectors where the heterogeneous meanings of a word or a sentence will be revealed according to the other words or sentences associated with it. Then, the network uses multi-scale CNN layers where filters are with multiple sizes to extract CNN features of a sentence or a paragraph from its attentive representations. Upon each CNN layer, the network further introduces a bidirectional-RNN layer with gated GRU to generate an RNN feature corresponding to the CNN features of a CNN layer. Finally, the network applies a fully-connected layer to merge the RNN features as the sentence- or paragraph- representation vectors.

The multi-scale CNN-RNN structure captures the coupling relations. Specifically, it captures the direct coupling relations between words or sentences by the CNN layers. The CNN layers model the direct coupling relations that are between a set of words or sentences with different sizes and different interactions by multiple filters with different sizes (i.e., the CNN filter 1 to CNN filter K in Figure 4.3). In this way, it obtains CNN features that are similar to the n-gram feature, which is widely used in traditional natural language processing methods. Different from the n-gram feature, which pays equal attention to every word- or sentence-sequence combination, the CNN features are more sensitive to the coupling relations with a high utility to the final sentiment analysis task (Bengio et al. 2013). The multi-scale CNN-RNN structure captures the indirect couplings by the bidirectional-RNN layers, which also reveal the sentiment polarity shift in a sentence or a paragraph with different scales. Consequently, the fully-connected layers integrate the direct and indirect coupling relations to reflect the comprehensive coupling relations in a sentence or a paragraph.

The attention mechanism captures the heterogeneous meanings of a word or a sentence in different language locales. This mechanism adjusts a representation vector of

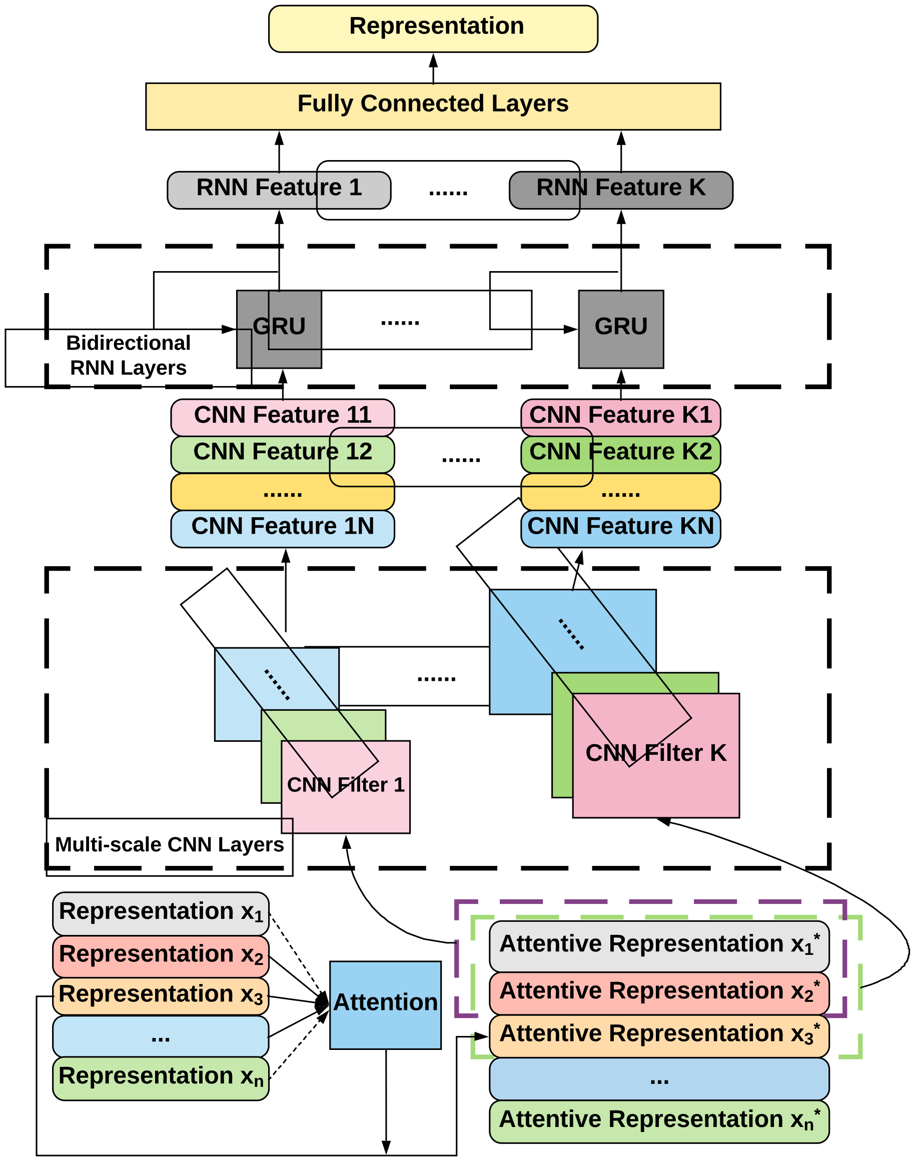


Figure 4.3: The Architecture of the non-IID-Characteristics-Learning Module in MEDEA.

a word or a sentence according to its context (i.e., the other inputted words or sentences associated with the word or the sentence). The adjusted representation vector reflects a specific meaning of the word or the sentence in a given context. As a result, it eliminates the sentiment ambiguities of the word or the sentence. For a representation vector x_i, the attention mechanism first uses a nonlinear layer to map it as $h_i = tanh(Wx_i + b)$, where W and b are the weight and bias of the nonlinear layer, respectively. Then, it calculates the adjusting factor α_i for the representation vector based on its context by a softmax function. The calculation of α_i is formalized in Equation (4.5).

$$\alpha_i = \frac{\exp(h_i^\top u)}{\sum_{j\in c(i)} \exp(h_j^\top u) + \exp(h_i^\top u)}, \tag{4.5}$$

where $\{x_j | j \in c(i)\}$ is the context set of x_i, and u is the global memory of the context that needs to be learned. According to the adjusting factor α_i, the attentive representation vector of x_i can be calculated as follows,

$$x_i^* = \alpha_i x_i. \tag{4.6}$$

The attentive representation vectors of the inputted words or sentences are further fed into the multi-scale CNN-RNN structure for coupling-relation learning.

The MEDEA method implements the non-IID-characteristics-learning module in both $E_s(\cdot)$ (the sentence-representation function) and $E_p(\cdot)$ (the paragraph-representation function) as the attentive multi-scale CNN-RNN structure. The rationale is that the non-IID characteristics at the word-level and the sentence-level have the same components and hierarchy (i.e., the direct and indirect couplings and heterogeneities) as analyzed in Chapter 4.2.1. Furthermore, a network with a designated structure can capture information with specific relations at different levels (e.g., the GoogLeNet hierarchically stacks the inception module to extract image features at different levels successfully; Szegedy et al. 2015). Accordingly, MEDEA can effectively capture the hierarchical non-IID characteristics in a non-IID paragraph through the stacking of the attentive multi-scale CNN-RNN structure.

4.3.2 Explicit Features used by the Multi-Scale and Hierarchical Deep Neural Network

To avoid over-fitting, the non-IID paragraph representation framework regulates the learning process with hierarchical sentiment-related explicit features. In this paper, MEDEA implements the explicit features at both word-level and paragraph-level.

4.3.2.1 Word-Level Explicit Features

The MEDEA method uses two word-level explicit features: (1) *sentiment lexicon* (Baccianella et al. 2010), and (2) *POS* (Xia & Zong 2011). To generate the sentiment lexicon feature, MEDEA encodes a word as a two-dimensional vector where the value in each dimension is an average of the positive or negative sentiment scores of the word. For a word that is not in the sentiment lexicon, MEDEA encodes it as $[0,0]$, assuming the word has the neutral sentiment polarity. To generate the POS feature, MEDEA encodes each word by a one-hot embedding where the value that is in the position corresponding to the word's POS is assigned as 1 while values that are in other positions are assigned as 0. The MEDEA method concatenates the sentiment-lexicon feature and the POS feature as the word-level explicit features. It then integrates the word-level explicit features with its implicit word representation by fully-connected layers.

The sentiment lexicon (a.k.a. sentiment dictionary) contains the probable sentiment polarity of each word. Typically, the sentiment lexicon represents the sentiment polarity of a word by sentiment scores per the positive and negative polarity, respectively. For example, the sentiment scores of the word "*suggestive*" in a sentiment dictionary (Baccianella et al. 2010) are 0.125 and 0.25 for the positive and negative polarities, respectively. Consequently, the MEDEA method uses $[0.125, 0.25]$ as the sentiment-lexicon feature of the word "*suggestive*". As a result, the sentiment-lexicon feature directly reveals the potential sentiment polarity of each word, which may contribute to the paragraph sentiment polarity.

The POS feature reflects the POS of each word. It explicitly points out the words that need to be paid more attention to sentiment analysis, because words with different POS may make different contributions to the sentiment polarity of a paragraph. It also explicitly indicates the heterogeneous meanings of a word because a word with different POS may have different meanings. For example, the POS of the word "*back*" in the sentence "*many of his friends backed his plan*" is a verb, and the meaning of the "*back*" here is support, which has positive sentiment polarity. On the contrary, the POS of the word "*back*" in the sentence *"she stumbled and fell, scraping her back badly"* is a noun, and "the *back*" has neutral sentiment polarity.

4.3.2.2 Paragraph-Level Explicit Features

The MEDEA method uses the *term-presentation* (Wiebe et al. 1999) as the paragraph-level feature. Specifically, MEDEA first counts the word-appearance frequency in the population. It then picks up 10% of the words with the lowest frequency as the rare-word set. Subsequently, it encodes the rare-word set by a one-hot embedding where the value that is in the position corresponding to an appeared word in the rare-word set will be set as 1, and the values that are in other positions will be set as 0.

The term-presentation feature captures rarely-appeared words. As demonstrated by Pang et al. (2002) and Taboada et al. (2011), some rarely-appeared words can directly determine the sentiment polarity of a paragraph. For example, the word “Goooooood" may only appear once or twice in a corpus; however, it likely indicates the positive sentiment polarity of a paragraph. As a result, the term-presentation feature reveals the significant words that have a high probability to determine the sentiment polarity for a paragraph. It should be noted that, when some significant rarely-appeared words exist in a paragraph, the other parts of the paragraph always make a limited contribution to the sentiment polarity. Because MEDEA combines the term-presentation feature with its implicit paragraph representation by fully-connected layers, in the MEDEA learning process, the backward-propagated errors may be passed to the explicit term-presentation feature instead of the complex neural networks when such rarely-appeared words exist. This error passing largely reduces the over-fitting risk of MEDEA.

4.4 Experiments and Evaluation of the Multi-Scale and Hierarchical Paragraph Representation Performance

This chapter evaluates the performance of MEDEA from four aspects:

1. *The sentiment analysis performance*: whether MEDEA can enable a better sentiment analysis performance.

2. *The sentiment embedding quality*: whether MEDEA can well embed the sentiment information and textual information into a representation space.

3. *The effectiveness of non-IID-characteristics learning*: whether MEDEA can effectively capture the non-IID characteristics in text.

4. *The significance of hierarchically integrating explicit features with implicit features*: whether the performance of MEDEA can be improved via hierarchically integrating explicit features with implicit features.

4.4.1 Data Sets for the Performance Evaluation of Non-Independent and Identically Distributed Representation Representation

The experiments verify the effectiveness of MEDEA on five large real-world data sets with sentiment labels, including the IMDB movie review (Diao et al. 2014), Amazon product review (Jindal & Liu 2008), and three data sets from *Yelp Challenges* (i.e., Yelp13, Yelp14, and Yelp15). The experiments also evaluate the MEDEA on two small real-world data sets from Twitter sentiment analysis task (Twitter) and Twitter airline sentiment analysis task (Twitter-Air). For IMDB, Yelp13, Yelp14, Yelp15, and Twitter data sets, the training and testing data sets have already been divided. For Twitter-Air and Amazon data sets, the experiments randomly split 90% and 10% data as the training and testing data, respectively.

Table 4.1 illustrates the statistical properties of these data sets. These data sets belong to several domains and applications, such as twitter: Twitter and Twitter-Air; movie: IMDB; recommendation platform: Yelp13, Yelp14 and Yelp15; and E-business website: Amazon. Their statistical properties are highly diversified, as indicated by data factors: the number of training documents ranges from 5,697 to 5,255,009, the average number of sentences ranges from 3 to 14, the average number of words ranges from 22 to 325.6, and the number of vocabularies ranges from 16,389 to 3,652,038. The data sets contain different numbers of sentiment classes with a maximum number of 10 classes. These diversified statistical properties indicate that these data sets may cover most circumstances in real-world paragraphs, which enables a fair environment for the evaluation.

Table 4.1: Statistical Properties of Data Sets: #s and #w refer to the number of sentences and words, respectively.

Data set	#train docs	#test docs	#class	average #s	max #s	average #w	max #w	#vocabulary
IMDB	280,593	34,029	10	14.0	148	325.6	2,802	115,831
Yelp13	268,013	33,504	5	9.0	154	143.8	1,176	188,434
Yelp14	900,363	112,549	5	9.2	151	156.9	1,199	476,191
Yelp15	1,255,409	156,928	5	9.0	151	151.9	1,199	612,636
Twitter	5,697	843	3	1.9	8	24.1	67	18,628
Twitter-Air	13,176	1,464	3	2.0	9	22.0	47	16,389
Amazon	5,255,009	583,889	9	9.2	5,424	189.0	7,094	3,652,038

4.4.2 Experimental Settings

4.4.2.1 Comparison Methods

The experiments compare MEDEA with four artificial features-based methods and eleven variants of seven state-of-the-art deep representation-based methods.

Most of these competitors are the same as that in Chapter 3.4. Two more advanced deep representation-based methods are compared in the experiments because these advanced methods also capture parts of non-IID characteristics. These two methods are briefly introduced as follows:

- ELMO (Peters et al. 2018): The ELMO method embeds the complex word relations and polysemy by leveraging the linguistic contexts and represents paragraphs based on the learned word embeddings through a bi-directional LSTM.
- BERT (Devlin et al. 2018): The BERT method adopts a bi-directional-training Transformer to capture complex word couplings in a paragraph. It inserts a special token at the beginning of a sentence and uses the output of its neural model corresponding to the token as a sentence representation vector.

These deep-representation-based methods act as competitors to evaluate whether MEDEA can capture non-IID characteristics and enable better sentiment analysis.

4.4.2.2 Data Preprocessing

In the pre-processing stage, MEDEA first splits a paragraph into several sentences and tokenizes these sentences by Stanford's CoreNLP toolkit (Manning et al. 2014). Then, it pre-trains the implicit word representation by the skip-gram model (Mikolov et al. 2013). Further, MEDEA annotates POS tags for each word using Stanford's CoreNLP and generates the sentiment lexicon feature based on the SentiWordNet (Baccianella et al. 2010) sentiment dictionary.

4.4.2.3 Neural-Network Implementation

The experiments set the default values of MEDEA's hyper-parameters as follows. The dimension of implicit word representation: 100. The filter size of the CNN in the multi-scale CNN layers: 2×100 and 3×100. The number of filters in each CNN layer in the non-IID-characteristics-learning module: 32 (at word-level) and 64 (at sentence-level). The number of GRU in the non-IID-characteristics-learning module: 64 (word-level) and 128 (sentence-level). The number of nodes in every fully-connected layer: 64 (at the word-level) and 128 (at the sentence-level). The number of layers of the fully-connected layers: 2. In the training phase, MEDEA uses a batch normalization after each layer and adopts a dropout strategy with keeping probability 0.5 after each fully-connected layer. It sets the batch size as 64 and uses the Adam algorithm (Kingma & Ba 2014) to optimize the learning-objective function.

The experiments set the parameters of the comparison methods as the same setting used in their original paper and fine-tune the pre-trained ELMO and BERT on Tensorflow Hub for each data set. For each competitor, the experiments feed its learned representation into a single-layer feed-forward neural network with the softmax function as the activation function in the output layer to constitute its sentiment classifiers.

4.4.3 Evaluation on Sentiment Classification Performance Enabled by the Non-Independent and Identically Distributed Representation Method

4.4.3.1 Evaluation Methods

The experiments test the sentiment classification performance of MEDEA to evaluate whether its captured non-IID characteristics can contribute to a better sentiment analysis result.

The performance of the sentiment classification is evaluated by two metrics: *accuracy* and *rooted-mean-square error (RMSE)*. Accuracy measures to what extent the enabled sentiment classification can assign the same rating as the ground truth. The higher accuracy refers to a better sentiment classification performance. Unlike typical classification tasks, in which a label contains nominal values, the labels in the sentiment classification task refer to sentiment ratings, which are a kind of ordinal value where a value has ordinal relation to others. For example, rating 4 is more similar to rating 5 than that to rating 1. In this case, a better sentiment classifier should predict a rating that is more similar to the rating of the ground truth. However, this property cannot be reflected by the accuracy metric. To complement the accuracy metric, RMSE is adapted to measure the similarity between the predicted and ground-truth rating. The smaller RMSE value indicates the better performance of a sentiment classifier.

4.4.3.2 Evaluation Results

The MEDEA method gains an accuracy improvement as much as 1.76%, as shown in Table 4.2. This accuracy improvement mainly benefits from (1) capturing the non-IID characteristics and (2) integrating explicit features with implicit features, which will be justified in Chapter 4.4.5. In Twitter and Twitter-Air data sets, MEDEA shows slightly worse performance compared with BERT but still achieves better performance than all other methods. The key reasons lie in that both of these data sets are with simple structures and relations, which are reflected by the small number of sentences and words in each paragraph as shown in Table 4.1, and they are only with very small data size (i.e., 5,695 and 13,176, respectively). For data sets with simple structures and relations, MEDEA does not have significant advantages because the non-IID characteristics in that data sets may not be significant. For data sets with small size, MEDEA is hard to fit well because of its large model complexity. Consequently, in these cases, the sentiment analysis performance enabled by MEDEA may be slightly worse than BERT, which has been pre-trained on huge data sets.

Table 4.2: Sentiment Classification Accuracy of Different Methods

Methods	IMDB	Yelp13	Yelp14	Yelp15	Twitter	Twitter-Air	Amazon
BOW	15.5	20.2	29.4	31.8	33.9	53.1	31.8
BOW-TFIDF	21.3	31.6	36.0	38.9	33.7	58.6	32.1
ngram	25.1	36.6	41.3	44.5	30.7	50.6	31.5
ngram-TFIDF	27.3	39.1	43.2	45.7	27.5	58.9	30.3
sentCNN-random	29.4	55.6	57.9	58.9	59.6	76.0	60.3
sentCNN-static	29.2	53.5	58.2	59.2	58.4	73.6	59.2
sentCNN-nonstatic	24.6	47.1	52.6	52.2	62.8	71.3	64.5
sentCNN-mul	29.4	53.3	58.3	58.7	57.7	75.8	61.3
gateRNN-CNN	40.6	61.4	62.7	63.2	52.8	61.2	56.2
gateRNN-LSTM	41.7	63.5	66.9	67.8	65.4	62.3	67.8
HNATT-ATT	42.5	63.2	67.3	68.1	64.8	78.2	69.8
HNATT-AVG	42.2	63.8	66.8	67.7	61.7	78.6	66.5
HNATT-MAX	42.9	63.7	66.8	68.0	65.4	79.3	68.2
ELMO	30.9	58.2	59.1	60.3	64.2	77.0	70.3
BERT	37.3	35.6	36.1	36.9	74.7	84.0	72.2
MEDEA	43.9	64.6	67.9	69.2	67.97	81.4	73.1

The MEDEA-enabled sentiment analysis also achieves the best performance in terms of RMSE in most of the data sets, as shown in Table 4.3. It always achieves the smallest RMSE except on the IMDB data set and improves as much as 5.26%. These results quantitatively justify that the MEDEA-enabled sentiment analysis performance is significantly better than that of its competitors. As can be seen in Table 4.3, the performance of MEDEA is better than that of the best state-of-the-art method, *HNATT-ATT*. Comparing with *HNATT-ATT*, which also adopts a hierarchical model and attention mechanism, MEDEA further reveals the non-IID characteristics at each level in a paragraph and hierarchically involves explicit features to better understand the characteristics of both the non-IID paragraph and the sentiment analysis task. As a result, MEDEA enables a better sentiment analysis performance.

Table 4.3: Sentiment Prediction RMSE of Different Methods

Methods	IMDB	Yelp13	Yelp14	Yelp15	Twitter	Twitter-Air	Amazon
BOW	3.0793	1.7230	1.5735	2.0959	1.1282	0.9949	2.9761
BOW-TFIDF	3.1717	1.6949	1.5425	1.7427	1.1340	0.9420	3.0152
ngram	2.7543	1.3934	1.2452	1.2567	1.1761	0.9303	2.8786
ngram-TFIDF	2.6471	1.3831	1.2331	1.251	1.2409	0.9599	2.6643
sentCNN-random	2.3654	1.0298	0.9686	0.9619	0.7655	0.6689	2.1246
sentCNN-static	2.5496	0.9894	0.9641	0.9466	0.8070	0.7309	2.0864
sentCNN-nonstatic	3.0942	1.3054	1.057	1.0951	0.7608	0.7615	1.9075
sentCNN-mul	2.5898	1.1216	0.9616	0.9375	0.8288	0.6591	1.9653
gateRNN-CNN	1.8956	0.8045	0.7892	0.7905	0.8801	0.9510	1.8022
gateRNN-LSTM	1.7497	0.7518	0.7108	0.7033	0.6837	0.8970	1.6437
HNATT-ATT	1.7256	0.7556	0.7017	0.688	0.7159	0.6407	1.5243
HNATT-AVG	1.7419	0.7536	0.7091	0.6923	0.7907	0.6135	1.6108
HNATT-MAX	1.7318	0.7553	0.6968	0.6895	0.7042	0.5937	1.5516
ELMO	2.3267	0.7738	0.9244	0.8768	0.7655	0.6533	1.5053
BERT	1.7087	1.7486	1.8118	1.8168	0.5875	0.5240	1.4692
MEDEA	1.7303	0.7267	0.6658	0.6532	0.6492	0.5543	1.4537

4.4.4 Evaluation on Representation Quality of the Non-Independent and Identically Distributed Representation Method

4.4.4.1 Evaluation Methods

The experiments further evaluate the sentiment embedding quality of MEDEA. The embedding quality is quantitatively measured by an information retrieval task and qualitatively illustrated by visualizing the representation. Different from the classification performance, the embedding quality reflects the generalization performance of MEDEA representation. A good sentiment embedding quality is essential for the success of a wide range of sentiment analysis tasks, such as sentiment classification and similar sentiment paragraph retrieval.

For the information retrieval task, the experiments use the Euclidean distance in a given paragraph-representation space to calculate the similarity of two paragraphs. For a given paragraph, the experiments retrieve its K-most-similar paragraphs (a.k.a., K-nearest paragraphs) under this similarity measurement. The retrieval performance is measured by RMSE@K. The RMSE@K metric calculates the RMSE between the sentiment ratings of a given paragraph and its K-nearest paragraphs. It reflects both local embedding quality (when K is small) and global embedding quality (when K is large). A lower RMSE@K indicates a higher sentiment embedding quality. In this experiment, the K used in RMSE@K is set as 1, 10, 100, and 1000.

4.4.4.2 Evaluation Results

The results of RMSE@K of each method on the Yelp13 data set are reported in Figure 4.4. The results show that MEDEA has the best sentiment embedding quality (improved up to 5.93% in terms of RMSE@K) in the information retrieval task, evidenced by having the smallest RMSE@K.

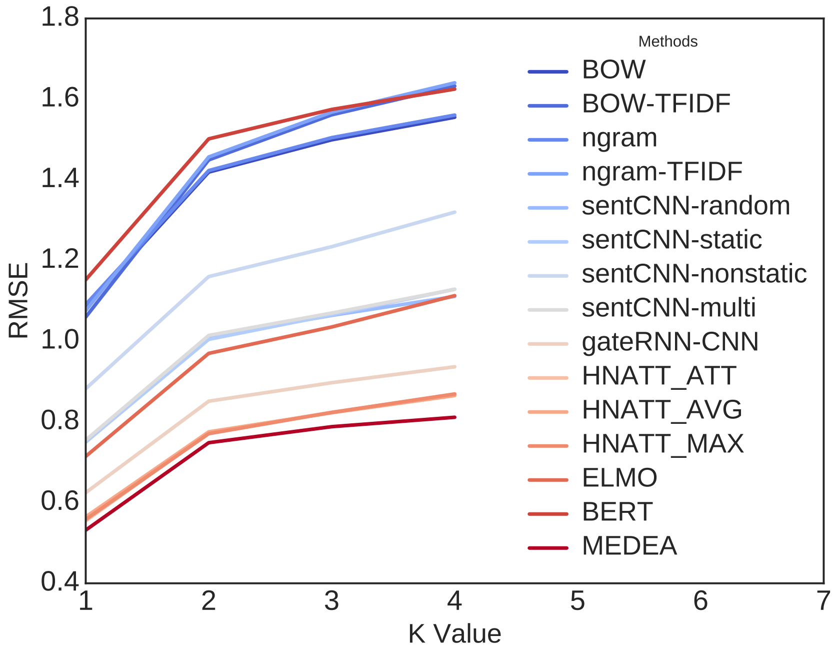
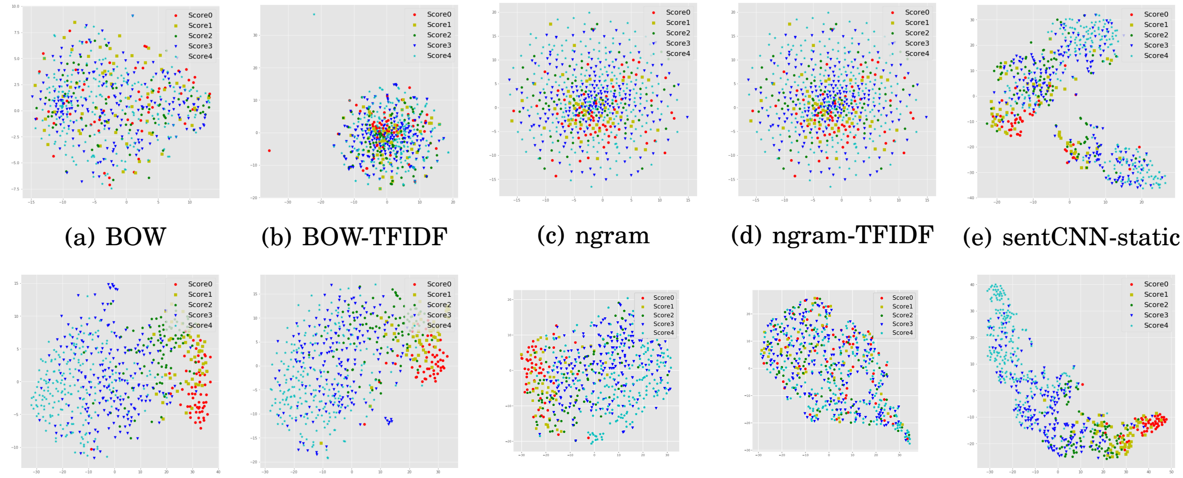


Figure 4.4: The RMSE@K of Different Methods on Yelp13.

To visualize the paragraph representation, this experiment transforms each representation vector from a high-dimensional representation space to a 2-dimensional vector space via the *t*-distributed stochastic neighbor embedding (t-SNE) (Maaten & Hinton 2008). It illustrates the transformed representation of MEDEA and the competitors on the Yelp13 data set in Figure 4.5[1]. As can be seen in Figure 4.5, the representation of MEDEA has smaller intra-class variances and larger inter-class variances compared with that of others. Meanwhile, the locations of paragraphs in the representation space are consistent with the order of their sentiment ratings. It demonstrates the sentiment embedding quality of MEDEA is better than others, which enables its superior sentiment analysis performance, as shown in Chapter 4.4.3.



(a) BOW (b) BOW-TFIDF (c) ngram (d) ngram-TFIDF (e) sentCNN-static

(f) gateRNN-LSTM (g) HNATT (h) ELMO (i) BERT (j) MEDEA

Figure 4.5: The Visualization of Different Representations through t-SNE Transformation on Yelp13.

4.4.5 Evaluation Significance of Hierarchically Integrating Explicit Features with Implicit Features

4.4.5.1 Evaluation Methods

This experiment evaluates the significance of hierarchically integrating explicit features with implicit features by comparing MEDEA with its two variants. The first variant is MEDEA-N, which adopts only the non-IID-characteristics-learning modules at different

[1] The experiment shows only one figure for one method due to space limitation.

levels to capture the non-IID characteristics (see Chapter 4.3.1). The second variant is MEDEA-W, which integrates only one explicit feature (i.e., POS) at word-level (see Chapter 4.3.2.1). To make a full comparison, this experiment evaluates these methods by all metrics used above (i.e., accuracy, RMSE, and RMSE@K). It further reports the accuracy of MEDEA-N and MEDEA on training and validation data set per epoch to demonstrate the effectiveness of reducing over-fitting by integrating explicit features.

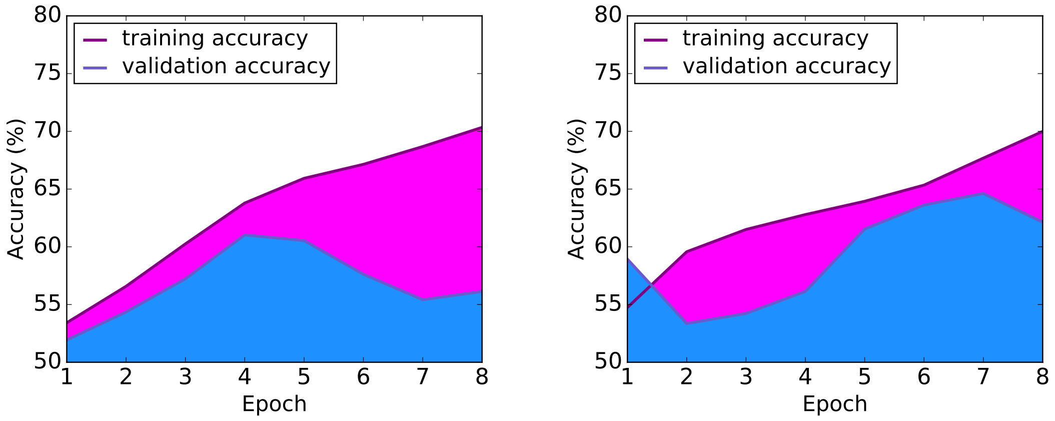
4.4.5.2 Evaluation Results

The accuracy, RMSE, and RMSE@K results are reported in Table 4.4. The results induce the following conclusions: (1) integrating explicit features indeed increases sentiment analysis performance, and (2) hierarchically integrating explicit features further improves sentiment analysis performance. The insight of the performance improvement is that MEDEA involves more sentiment-related information, which preserves the distribution properties of a paragraph space in its representation space, compared with its two variants.

Table 4.4: Sentiment Analysis Performance based on MEDEA and Its Variants

Criteria	Methods	IMDB	Yelp13	Yelp14	Yelp15
Accuracy	MEDEA-N	40.7	61	64.4	64.6
	MEDEA-W	43.1	64	67.2	68.6
	MEDEA	**43.9**	**64.6**	**67.9**	**69.2**
RMSE	MEDEA-N	1.9602	0.7821	0.8134	0.7797
	MEDEA-W	1.7316	0.7427	0.6872	0.6602
	MEDEA	**1.7303**	**0.7267**	**0.6658**	**0.6532**
RMSE@K	MEDEA-N	0.5988	0.8157	0.8532	0.8842
	MEDEA-W	0.5433	0.7695	0.8025	0.8401
	MEDEA	**0.5318**	**0.7485**	**0.7882**	**0.8115**

The training and validation accuracy of MEDEA on Yelp13 is illustrated in Figure 4.6. Compared with MEDEA-N, the validation accuracy of MEDEA is much higher and more stable, while the training accuracy is increasing. On the contrary, the validation accuracy of MEDEA-N drops rapidly after a few steps. These results demonstrate that the complex network structure of MEDEA faces over-fitting, and integrating the explicit features reduces the over-fitting effectively.



(a) MEDEA-N

(b) MEDEA

Figure 4.6: The Training and Validation Accuracy of MEDEA on Data Set Yelp13.

4.5 Summary

This chapter proposes a non-IID paragraph representation framework to embed the complex hierarchical coupling relations and heterogeneous meanings of words and sentences into a vector space for sentiment analysis. It instantiates this framework as a multi-scale and hierarchical deep neural network with an attention mechanism, namely MEDEA. The MEDEA method learns the non-IID characteristics by a multi-scale CNN-RNN structure and hierarchically combines the learned implicit features with the artificially extracted explicit features, including the sentiment lexicon feature, the POS feature, and the term-presentation feature. The comprehensive experimental results support the effectiveness of non-IID-characteristics learning and demonstrate the superior sentiment analysis performance enabled by MEDEA.

MEDEA represents a paragraph by comprehensively and hierarchically modeling its non-IID paragraph characteristics (i.e., couplings and heterogeneities) within and between words and sentences and by integrating the explicit features with implicit features. As a result, MEDEA enables significantly better sentiment analysis performance for paragraphs with strong non-IID characteristics but slight improvements for paragraphs with simple structures and relations. Furthermore, MEDEA gains excellent benefits from integrating explicit features that reflect domain knowledge. However, how to effectively identify and extract useful explicit features for a specific data set is still an open problem.

Chapter 5

Sentiment Analysis on Short Text

5.1 Introduction

Social media data contains a large number of low quality data, which are very short and always with informal words and typos. Effectively representing short texts into a vector space that embeds the semantic meaning of the texts is valuable and required by reliable sentiment analysis. Different from the Chapters 3 and 4, which focus on formal long text (paragraph) representation, this chapter studies short-text representation for sentiment analysis on short text.

Short-text representation is very challenging compared with formal long-text representation, which studied in Chapters 3 and 4, because of its two essential characteristics: *noise* (Dey et al. 2016) and *sparsity* (Wang, Wang, Zhang & Yan 2017), where *noise* refers to informal words and typos, and *sparsity* means the rare number of vocabularies in a text (because of the short length limitation). As a result, most of the current text representation methods may fail to represent short text. For example, the word2vec-based methods, such as the methods in (Kim 2014, Tang, Qin & Liu 2015), need to look up pre-trained word representations, but the informal words and typos may not appear in the training data. For another example, when facing sparsity, the term-frequency-inverse-document-frequency (TF-IDF) and the bag-of-words method generate a very sparse representation for a short text (i.e., most of the entries in a representation vector are 0), which may be meaningless for downstream learning tasks because distances between all texts are equal.

Recently, several methods have been proposed for short text representation. They achieved advanced performance by either reducing noise or alleviating sparsity. Regarding noise reduction, most of the current methods (Dey et al. 2016, De Boom et al. 2016, Wang, Wang, Zhang & Yan 2017, Arora et al. 2017*a*, Li, Wang, Zhang, Li, Chi & Ouyang 2018) first recognize noise by looking up pre-defined noise sets or adopting frequency-based detection models. Then, they reduce the effects of the recognized noise by re-weighting or ensemble strategies. However, two problems may arise in their recognition process: (1) pre-defined noise sets may not fully cover all noise; and (2) frequency-based detection models may fail when facing the sparsity. As a result, their unrecognized noises may still damage representation performance, even with re-weighting and ensemble strategies. Regarding sparsity alleviation, most of the current methods (Zuo et al. 2016, Liang et al. 2016, Lochter et al. 2016, Li, Li, Chi & Ouyang 2018) assume a sparse short text is generated from a latent dense document, and try to insert words into the short text according to the latent document, which is also known as the expansion-based method. However, the quality of the expansion cannot be guaranteed because (1) many short texts are independent, that is they are not generated from the same document; and (2) most of these methods are based on statistics which may profoundly be affected by the noise in a short text.

This chapter tackles the above problems in low-quality data representation by learning *multi-grain noise-tolerant patterns* in texts and embedding the most significant patterns in a text into a dense vector space to represent the low-quality text. Here, the noise-tolerant pattern means the textual pattern whose meaning is not affected by noise in a short text. The motivation behind the multi-grain noise-tolerant patterns learning is that no matter how many noises in texts and how sparse the texts are, there should have one or more noise-tolerant patterns that exist in different texts with the same semantic meaning. For example, two sentences “Does anyone know how to repair?:(" and “dz ne1 knw h2 ripair?:(" have the same meaning. Although the second sentence has many informal words and typos, it has many explicit noise-tolerant patterns, which are the same as that in the first sentence, such as “knw" and “rpair? at character-level and “:(" symbol at word-level. These noise-tolerant patterns reflect the semantic meaning of the short texts. Effectively embedding such patterns into a dense space can avoid the meaningless sparse representation caused by text sparsity.

This chapter embeds the multi-grain noise-tolerant patterns by a bi-level neural network to capture the semantic relations among words and also among characters with different granularities to tackle the sparsity problem. The intuition is that the semantic meaning of a short text can be reflected by the relations among words and also among characters with different granularities, as shown in the above example.

To further tolerant noise, this chapter proposes a *breaking-gathering* strategy. In the breaking stage, the strategy breaks a piece of text into all combinations of its vocabularies but keeps their ordinal information in the text. Then, at the gathering stage, it discovers the most significant patterns in these combinations as the pattern of the piece of text. Through this process, the breaking-gathering strategy can adaptively filter noise with arbitrary form, and thus discover the noise-tolerant patterns. For example, at the character-level, the word “know" is broken as “know", “kow", “knw", “kw" and so on in the breaking stage. Considering “knw" in another short text, the noise-tolerant pattern “knw" can be discovered in the gathering stage.

Based on the above analysis, this chapter proposes a **bi**-level **m**asked multi-sc**a**le **c**onvolutional and **r**ecurrent neural netw**o**rk (Bi-MACRO). The Bi-MACRO method jointly captures multi-gran relations of words and characteristics to discover noise-tolerant patterns and embeds them as a dense vector representation for a short text.

The key contributions of this chapter include the following:

- *A bi-level neural network representation architecture.* The bi-level neural network representation architecture captures the semantic meaning of a text from both word-level and character-level, and it embeds the captured semantic meaning into a dense vector space that tackles the sparsity problem.

- *A masked CNN layer for adaptively noise filtering.* The masked CNN layer filters noise by a set of masks, and adaptively selects the most significant pattern by a cross-filter max pooling. Combining with the bi-level architecture, the masked CNN layer significantly reduces the noises at both word-level and character-level.

- *A multi-scale CNN-RNN structure.* Bi-MACRO uses a set of CNN with different filter sizes to capture short-term word and character relations with different granularities. The different filter sizes also induce masks with different mask locations, which fit the noise with an arbitrary position. Further, the connected RNN layer captures the long-term relations between words and characters, and it reduces the potential model complexity that may be caused by adopting a large filter size.

This chapter conducts comprehensive experiments on five widely used real-world data sets, including TREC, Quora, Twitter, News, and AG News, to show the short-text characteristics and evaluate the performance of the proposed method. The experimental results show that the proposed Bi-MACRO method significantly outperforms three state-of-the-art competitors and two baseline methods in terms of short-text representation.

The rest of this chapter is organized as follows. Section 5.2 introduces the proposed method. Section 5.3 demonstrates the Bi-MACRO performance by comparing it with the state-of-the-art short-text representation methods. Lastly, Section 5.3.6 concludes this chapter.

5.2 Bi-level Masked Multi-Scale Convolutional-Recurrent Neural Network for Short Text Sentiment Analysis

5.2.1 The Architectural of Bi-level Masked Multi-Scale Convolutional-Recurrent Neural Network

The architecture of Bi-MACRO is shown in Figure 5.1. Given a short text, Bi-MACRO first represents it into a word embedding matrix and a character embedding matrix. Then, for each matrix, Bi-MACRO adopts a masked multi-scale CNN-RNN network to learn a vector representation. Finally, Bi-MACRO integrates these two vectors to a unified vector as the short-text representation.

To tackle the sparsity problem, Bi-MACRO embeds multi-granularity relations among both words and characters by a bi-level (i.e., character-level and word-level) multi-scale CNN and RNN structure. In this way, the word-level structure can process formal words, while the character-level structure can handle informal words. To further tolerant noises caused by informal words and typos, Bi-MACRO implements the *breaking-gathering* strategy by masked CNN layers, which will be introduced in Section 5.2.4.

5.2.2 The Transformation of Short Textual Data

Given a short text $T = \{t_1, t_2, \cdots, t_{n_w}\}$, Bi-MACRO transforms it into a word embedding matrix $\mathbf{E}_w \in \mathbb{R}^{n_w \times n_e^w}$ and a character embedding matrix $\mathbf{E}_c \in \mathbb{R}^{n_c \times n_e^c}$ by looking up the transformation matrices $\mathbf{T}_w \in \mathbb{R}^{n_W \times n_e^w}$ and $\mathbf{T}_c \in \mathbb{R}^{n_C \times n_e^c}$, where n_w corresponds to the maximum number of words in a short text, n_c corresponds to the maximum number of characters in a short text, n_W refers to the number of unique vocabularies in the corpus, n_C refers to the number of unique characters in the corpus, n_e^w and n_e^c refers to the

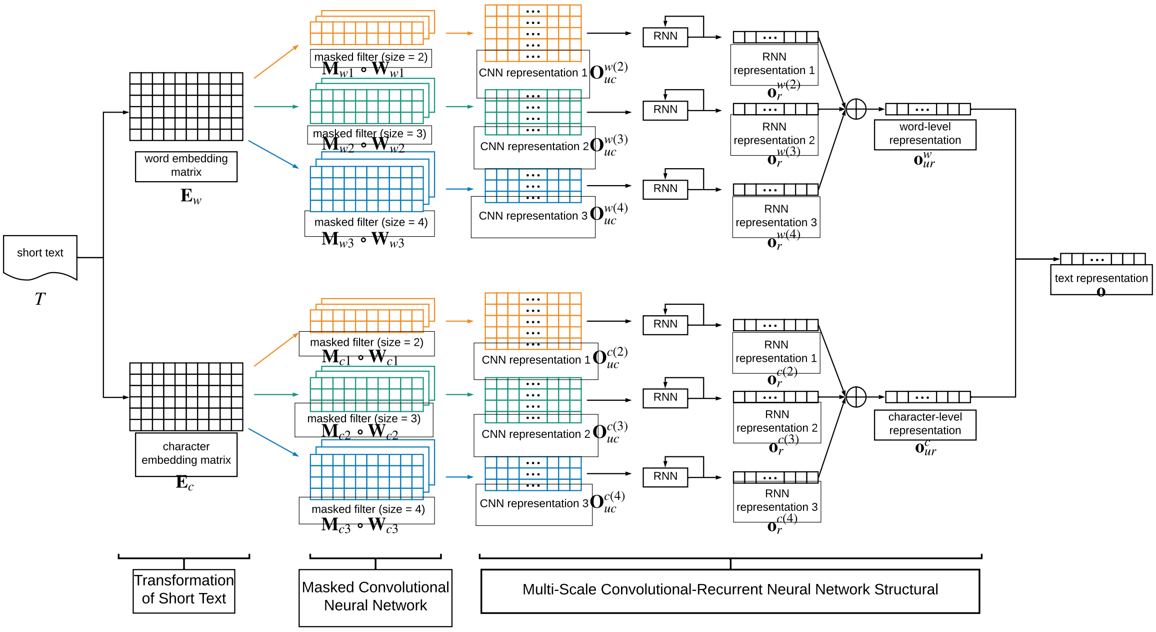

short text
T
word embedding matrix
E_w
character embedding matrix
E_c
masked filter (size = 2)
M_w1 ∘ W_w1
masked filter (size = 3)
M_w2 ∘ W_w2
masked filter (size = 4)
M_w3 ∘ W_w3
masked filter (size = 2)
M_c1 ∘ W_c1
masked filter (size = 3)
M_c2 ∘ W_c2
masked filter (size = 4)
M_c3 ∘ W_c3
CNN representation 1 O_uc^w(2)
CNN representation 2 O_uc^w(3)
CNN representation 3 O_uc^w(4)
CNN representation 1 O_uc^c(2)
CNN representation 2 O_uc^c(3)
CNN representation 3 O_uc^c(4)
RNN
RNN representation 1 o_r^w(2)
RNN representation 2 o_r^w(3)
RNN representation 3 o_r^w(4)
RNN representation 1 o_r^c(2)
RNN representation 2 o_r^c(3)
RNN representation 3 o_r^c(4)
word-level representation o_ur^w
character-level representation o_ur^c
text representation o
Transformation of Short Text
Masked Convolutional Neural Network
Multi-Scale Convolutional-Recurrent Neural Network Structural

Figure 5.1: The Architecture of Bi-MACRO.

dimension of the word and character embedding, respectively. Initially, to leverage the word semantic meaning, Bi-MACRO adopts a pre-trained word embedding matrix as $\mathbf{T}_w$ (e.g., the embedding matrix pre-trained by GloVe algorithm [1]; Pennington et al. 2014) and randomly generates a matrix as $\mathbf{T}_c$. Then, it optimizes $\mathbf{T}_w$ and $\mathbf{T}_c$ in its learning process.

5.2.3 Masked Convolutional Neural Network

The proposed masked convolutional network is shown in Figure 5.2. For a CNN filter, this thesis denotes its weight matrix as $\mathbf{W} \in \mathbb{R}^{n_{f_h} \times n_{f_w}}$ where n_{f_h} and n_{f_w} are the height and width of the filter, respectively. The Bi-MACRO network sets the filter width n_{f_w} as the dimension of word embedding n_{ew} at the word-level, and it sets n_{f_w} as the dimension of character embedding n_{ec} at the character-level, considering the spatial relation of a text matrix is among words and characters instead of embedding features. The Bi-MACRO network masks the weight matrix $\mathbf{W}$ by entry-wise multiplying the weight matrix with a mask matrix $\mathbf{M} \in \{0,1\}^{n_{f_h} \times n_{f_w}}$ (a position of 0 in $\mathbf{M}$ will mask the input of the corresponding position). Formally, given a word or character embedding matrix $\mathbf{E} \in \mathbb{R}^{n \times n_e}$, Bi-MACRO calculates the output of a masked CNN filter (a CNN filter with a

[1] The pre-trained word embedding can be downloaded from http://nlp.stanford.edu/data/glove.6B.zip

masked weight matrix) as

$$\mathbf{o}_{mc} = [o_1, o_2, \cdots, o_{n-n_{f_h}+1}]^\top. \tag{5.1}$$

In Eq. (5.1), the k-th entry of $\mathbf{o}_{mc}$ is

$$o_k = g(\sum_{i=1}^{n_{fw}} \sum_{j=1}^{n_e} \mathbf{M}_{i,j}\mathbf{W}_{i,j}\mathbf{E}_{k+i-1,j} + b), \tag{5.2}$$

were $g(\cdot): \mathbb{R} \to \mathbb{R}$ is a non-linear function, and $b \in \mathbb{R}$ is a bias term. This thesis uses *ReLU* as the non-linear function $g(\cdot)$ in each masked CNN. As demonstrated in Figure 5.2, a masked CNN has many filters with the same size to capture different relations among words or characters with the same granularity. Accordingly, for these filters and a set of mask matrices with same entry values, a masked CNN calculates a set of output vectors, and it stacks these vectors as the output matrix:

$$\mathbf{O}_{mc} = [\mathbf{o}_{mc_1}, \mathbf{o}_{mc_2}, \cdots, \mathbf{o}_{mc_{n_f}}]^\top, \tag{5.3}$$

where n_f refers to the number of filters. The $\mathbf{O}_{mc}$ is also known as the CNN features.

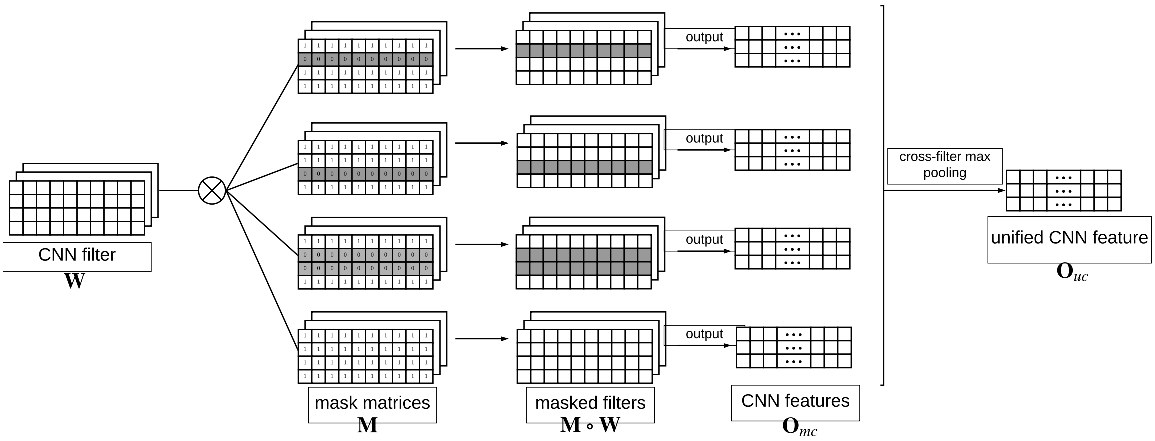


Figure 5.2: Masked Convolutional Network. The shadow parts refer to masks.

Because the noise may appear at any position between normal words and characters, masks with different mask locations should be adopted. To tackle arbitrary noise positions, masked CNN masks every combination of rows between the first and the last rows for a filter, as shown in Figure 5.2. Specifically, masked CNN generates $2^{(n_{f_h}-2)}$ different mask matrices for a filter with n_{f_h} height. As a result, it will have $2^{(n_{f_h}-2)}$ CNN feature matrices,

$$O_{mc} = \{\mathbf{O}_{mc}^{(1)}, \mathbf{O}_{mc}^{(2)}, \cdots, \mathbf{O}_{mc}^{2^{(n_{f_h}-2)}}\}. \tag{5.4}$$

Finally, masked CNN adopts a cross-filter max pooling to integrate its CNN feature matrices as a unified CNN feature matrix. Here, the cross-filter max-pooling compares the values at the same entry in different CNN feature matrices and assigns the max value as the value at the same entry in the unified CNN feature matrix. Formally, the value at the (i,j)-th entry in the unified CNN feature matrix can be calculated as:

$$o_{uc_{ij}} = \max(o^{(1)}_{mc_{ij}}, o^{(2)}_{mc_{ij}}, \cdots, o^{2^{(n_{f_h}-2)}}_{mc_{ij}}), \tag{5.5}$$

where $o^{(k)}_{mc_{ij}}$ is the value at the (i,j)-th entry of the k-th CNN feature matrix in O_{mc}. The rationale is that each entry corresponding to a specific pattern between words or characters, and the largest value corresponding to the most significant pattern. By adopting the cross-filter max pooling, masked CNN can always extract the most significant pattern in text that is tolerant to noise.

5.2.4 Multi-Scale Convolutional-Recurrent Neural Network Structural

In order to capture multi-grain patterns, Bi-MACRO adopts CNN with multi-scale filter sizes. It should be noted that the introduced $2^{(n_{f_h}-2)}$ masks may dramatically increase the model complexity when n_{f_h} is large. Such high model complexity will cause the model learning intractable. To reduce the model complexity, this thesis only uses the filter with size $2,3,4$, which only increases extra mask matrices by $\frac{4}{3}$ times. With these small size filters, masked CNN can capture local noise-tolerant patterns but fail to capture the long-term global relations among words or characters. To fill this gap, Bi-MARCO introduces a recurrent neural network (RNN) after each masked CNN to leverage such long-term global relations. This thesis adopts the gated recurrent unit (GRU) to implement the RNN. Specifically, each row in CNN feature matrix $\mathbf{O}_{uc}$ is fed into a GRU sequentially. For the t-th row in $\mathbf{O}_{uc}$, the output $\mathbf{h}_t \in \mathbb{R}^{1\times n^r_e}$ of the GRU is computed as follows:

$$\begin{aligned}
\mathbf{z}_t &= \sigma(\mathbf{O}_{uc_t}\mathbf{U}^z + \mathbf{h}_{t-1}\mathbf{V}^z),\\
\mathbf{r}_t &= \sigma(\mathbf{O}_{uc_t}\mathbf{U}^r + \mathbf{h}_{t-1}\mathbf{V}^r),\\
\hat{\mathbf{h}}_t &= \tanh(\mathbf{O}_{uc_t}\mathbf{U}^h + (\mathbf{r}_t \cdot h_{t-1})\mathbf{V}^h),\\
\mathbf{h}_t &= (1-\mathbf{z}_t)\cdot\mathbf{h}_{t-1} + \mathbf{z}_t\cdot\hat{\mathbf{h}}_t,
\end{aligned} \tag{5.6}$$

where n^r_e is the dimension of the RNN embedding, $\sigma(\cdot): \mathbb{R}^{1\times n^r_e} \to \mathbb{R}^{1\times n^r_e}$ is the sigmoid function, $\tanh(\cdot): \mathbb{R}^{1\times n^r_e} \to \mathbb{R}^{1\times n^r_e}$ is the tanh function, $\mathbf{r} \in \mathbb{R}^{1\times n^r_e}$ is a reset gate, $\mathbf{z} \in \mathbb{R}^{1\times n^r_e}$ is an update gate, and $\mathbf{U}^z$, $\mathbf{U}^r$, $\mathbf{U}^h$, $\mathbf{V}^z$, $\mathbf{V}^r$, and $\mathbf{V}^h \in \mathbb{R}^{(n_{f_h}-2)^2\times n^r_e}$ are the transform matrices in the GRU. The Bi-MACRO network uses the last output $\mathbf{h}_{(n_{f_h}-2)^2}$ of the GRU as the RNN output $\mathbf{o}_r$ of a masked CNN.

For masked CNN with filter sizes 2,3,4, Bi-MACRO generates three RNN outputs $\mathbf{o}_r^{(2)}$, $\mathbf{o}_r^{(3)}$ and $\mathbf{o}_r^{(4)}$. Similar to the cross-filter max-pooling in the masked CNN, Bi-MACRO here adopts a cross-RNN max pooling to integrate these RNN outputs. Formally, the i-th entry in the unified RNN output $\mathbf{o}_{ur}$ is calculated as:

$$o_{ur_i} = \max(o_{r_i}^{(2)}, o_{r_i}^{(3)}, o_{r_i}^{(4)}). \tag{5.7}$$

Finally, Bi-MACRO concatenates the unified RNN outputs at the word-level and character-level to form the short-text representation:

$$\mathbf{o} = [\mathbf{o}_{ur}^{w\top}, \mathbf{o}_{ur}^{c\top}]^\top. \tag{5.8}$$

The short-text representation $\mathbf{o}$ is then fed into a downstream text analytic tasks such as text category classification and sentiment classification. The Bi-MACRO network is jointly trained with the downstream task to represent short text in an end-to-end fashion.

5.3 Experiments and Evaluation of Sentiment Analysis on Short Text

This chapter evaluates the performance of Bi-MACRO from three aspects:

1. *The short text classification performance*: whether Bi-MACRO can enable a more accurate sentiment classification on the short text.

2. *The short text retrieval performance*: whether Bi-MACRO can enable a more precise short text retrieval, which is another essential task for sentiment analysis (a customer always confirm the reliability of a review by checking other reviews that have the same semantic meaning and sentiment polarity as the review).

3. *The short text representation quality*: whether Bi-MACRO can well embed the short text information into a representation space. High representation quality is the foundation for the downstream sentiment analysis tasks, such as sentiment clustering.

5.3.1 Data Sets

The experiments are conducted on five widely used real-world short-text data sets. These data sets include question answering data sets: TREC[2], Quora[3]; social media data sets: Twitter[4]; article title data sets: News[5], AG News[6]. For AG News data set, we select five comparable categories: entertainment, sports, business, sci/tech, and health. For AG News and Quora data sets, we randomly use 95% and 5% objects in a data set as the training and testing data, respectively. For other data sets, we use the originally provided training and testing sets. The characteristics of each data set are shown in Table 5.1. As shown in the Table 5.1, the noise and sparsity appear in all data sets. Data set with a larger ratio of unknown words represents more noise, and data set with shorter length shows a larger degree of sparsity.

Table 5.1: The Data Characteristics of Each Short-Text Data Set.

Data Set	TREC	Quora	Twitter	News	AG News
#Texts	10,764	537,933	11,394	20,120	471,542
#Class	6	2	3	8	5
#Voc.	8,872	105,929	20,587	24,201	70,592
#Unk Voc.	3,774	48,699	13,782	7,936	51,773
Avg. Len.	6.10	12.97	12.53	9.44	4.53
Avg. Char.	29.09	61.86	59.98	72.39	21.28

5.3.2 Experimental Settings

5.3.2.1 Comparison Methods

The experiments compare Bi-MACRO with three state-of-the-art short-text representation methods and two baseline methods to evaluate Bi-MACRO's performance. The following part briefly summarizes the state-of-the-art competitors:

- CNN (Zhang et al. 2015): CNN has been adopted at the character level to represent text, which can relieve the effects of ad-hoc symbols in short text.

[2] http://cogcomp.cs.illinois.edu/Data/QA/QC

[3] https://www.kaggle.com/c/quora-question-pairs/data

[4] https://www.cs.york.ac.uk/semeval-2013/task2.html

[5] http://acube.di.unipi.it/tmn-dataset/

[6] http://www.di.unipi.it/ gulli/AG_corpus_of_news_articles.html

- WWE (De Boom et al. 2016): this method learns a weighted aggregation of word representations to represent short text where the loss function is customized to reduce the effects of noise.
- SIF (Arora et al. 2017*a*): this method also adopts word embedding aggregation, but it holds an assumption that some words occurring out of context, which reduces the hazards brought by the sparsity.
- SeaNMF (Shi et al. 2018): this method leverages the local word-context relation to enhancing a non-negative matrix factorization method, catering for the sparsity challenge of the short-text representation.

The experiments select the TF-IDF and LDA (Blei et al. 2003) as the baseline methods to demonstrate the effects of the short-text characteristics (i.e., *noise* and *sparsity*) to typical text representation methods.

5.3.2.2 Neural-Network Implementation

For Bi-MACRO, the experiments empirically set the number of filters with the same size as 100, the number of RNN units as 100, the word-level and character-level text embedding dimension as 100. The experiments use *ReLU* function as the activation function in each hidden unit in Bi-MACRO, and adopt *Adam* (Kingma & Ba 2014) as the optimization method to train the Bi-MACRO with batch size 32. For the competitors, the experiments adopt their default setting reported in their corresponding paper. The experiments use the pre-trained word embedding by the GloVe algorithm (Pennington et al. 2014) in Bi-MACRO, WWE, and SIF; the experiments randomly initialize the embeddings of unknown words.

5.3.3 Evaluation on Bi-level Masked Multi-Scale Convolutional-Recurrent Neural Network Performance on Short Text Classification

This experiment adopts different classification mechanisms on the Quora data set and the other data sets. Because the label of the Quora data set is whether two sentences are the same question (i.e., with the same meaning), and the label of the other data sets are the text categories.

For the Quora data set, this experiment first adopts each representation method to represent the sentences, and then, it concatenates the representations of two sentences as the input of a classifier to classify whether they are the same question. This experiment uses a three-layers fully-connected neural network as the classifier. It sets the number of hidden units in each hidden layer of the classifier as 100 and uses *tanh* as the activation function in each hidden unit. For the other data sets, this experiment feeds the representation of each text into a three-layers fully-connected neural network with the same setting as that for the Quora data set.

The accuracy of the short-text classification enabled by different methods is reported in Table 5.2. As can be seen in Table 5.2, Bi-MACRO enables the best performance in all data sets. In Table 5.2, the performance improvement ratio (Δ) of Bi-MACRO compared with the other method with the highest accuracy is also reported to demonstrate the significance of Bi-MACRO. In this experiment, Bi-MACRO improves up to 4.87% on the AG News data set, which has the most significant noise and sparsity, as shown in Table 5.1. This result not only illustrates Bi-MACRO significantly improves the short-text classification performance but also demonstrates that Bi-MACRO effectively tackles the noise and sparsity in short-text representation.

Table 5.2: The Short Text Classification Performance based on Different Representation Methods.

Data Set	TREC	Quora	Twitter	News	AG News
TF-IDF	96.80	78.62	59.67	63.90	71.44
LDA	77.20	74.17	49.70	43.86	44.98
WWE	96.80	78.94	44.96	15.44	61.11
SeaNMF	25.40	61.87	48.75	16.84	22.11
SIF	95.80	77.98	60.26	77.89	75.37
Bi-MACRO	**98.00**	**81.61**	**61.45**	**79.56**	**79.04**
Δ	1.24%	3.38%	1.97%	2.14%	4.87%

5.3.4 Evaluation on Bi-level Masked Multi-Scale Convolutional-Recurrent Neural Network Performance on Short Text Retrieval

This experiment further evaluates the Bi-MACRO representation performance through short-text retrieval. It uses the short texts in the testing set as queries, and it reports the precision@k (i.e., the fraction of k-closest short texts selected per the Euclidean distance in a representation space that are the same-class neighbors) as the metric of the text retrieval performance. The value of k is set as 5, 10, and 20 in the experiment. This experiment conducts short-text retrieval on the AG News data set because it has the most significant noise and sparsity.

The results of short-text retrieval in terms of precision@k are reported in Table 5.3. These results demonstrate that Bi-MACRO significantly improves the performance of short-text retrieval (up to 39.34%) compared with state-of-the-art methods. Furthermore, with the number of retrieval texts (k) increasing, the precision@k of all methods instead of Bi-MACRO decreases rapidly. This result reflects that Bi-MACRO captures much more noise-tolerant patterns in multiple granularities; these captured multiple-granularity patterns preserve the short-text local distribution.

Table 5.3: The Short Text Retrieval Performance based on Different Representation Methods.

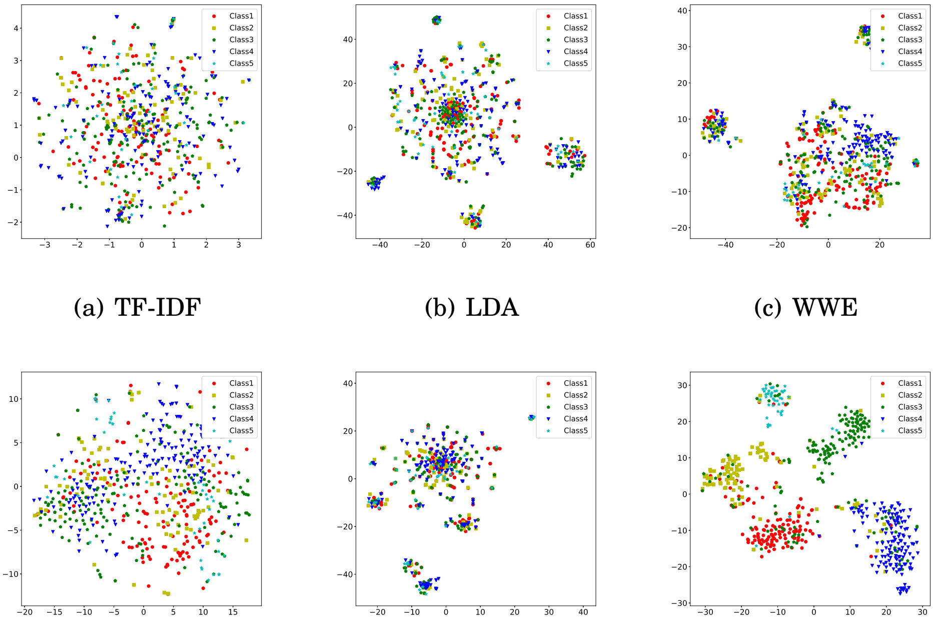
Metric	Precision@5	Precision@10	Precision@20
TF-IDF	42.65	40.50	38.31
LDA	34.23	32.51	31.07
WWE	45.49	44.12	42.70
SeaNMF	44.36	41.95	40.69
SIF	55.04	52.83	50.48
Bi-MACRO	**70.76**	**70.56**	**70.34**
Δ	28.56%	33.56%	39.34%

5.3.5 Evaluation on Short Text Representation Quality of Bi-level Masked Multi-Scale Convolutional-Recurrent Neural Network

This experiment visualizes the short-text representation on the AG News testing data set in a two-dimensional space trough TSNE (Maaten & Hinton 2008). As shown in Figure 5.3, it plots the category label of each short text in different colors to evaluate the representation quality; a high-quality short-text representation will clearly separate texts in different categories in the representation space.

In the Bi-MACRO generated representation space, the short texts in the same category are clearly clustered together. In contrast, the representation of other methods mixes texts with different categories. The rationale is that Bi-MACRO captures the multi-grain noise-tolerant patterns in short text by the bi-level masked multi-scale

CNN-RNN structure, which significantly filters the noise and fits the sparse text. As a result, the semantic meaning of a short text is propriety reflected by Bi-MACRO's representation.



(a) TF-IDF

(b) LDA

(c) WWE

(d) SIF

(e) SeaNMF

(f) BiMACRO

Figure 5.3: Short Text Representation of Different Methods.

5.3.6 Summary

This chapter proposes a bi-level masked multi-scale CNN-RNN networks to tackle the noise and sparsity problems in short-text representation. The proposed representation method learns multi-grain noise-tolerant patterns and then embeds the most significant patterns in a short text as its representation. It can effectively represent short text and significantly improves the downstream analytic tasks, as demonstrated by comprehensive experiments.

CHAPTER 6

FRAUDULENT SENTIMENT ANALYSIS

6.1 Introduction

Social media is becoming increasingly significant and profoundly affects our daily life. Unfortunately, a large proportion of social media reviews are proposed by fraudsters for strong incentives of profit and reputation. For example, 25% of reviews on Yelp (a social media-based recommendation website) might be a fraud, as reported in 2013[1]. This proportion has rapidly increased as observed in 2017[2]. These fraudulent reviews mislead general sentiment analysis methods (e.g., the method proposed in Chapters 3, 4, and 5) to analyze sentiment correctly. As a result, effectively detecting such fraudulent sentiment is a critical task that has excellent business values. To complement the research in the previous chapters, this chapter thus focuses on fraudulent sentiment analysis.

As discussed in Section 2.3, current efforts on fraudulent sentiment analysis mainly focus on analyzing *statistical information on historical reviewing activities* (e.g., the characteristics a user always displays when this user is writing a review) and *link-based social relations* (e.g., user-user relation, user-item relation, and item-item relation in social media) (Ye & Akoglu 2015, Rayana & Akoglu 2015, Hooi et al. 2017, Liu et al.

[1] https://www.bbc.com/news/technology-24299742

[2] https://www.forbes.com/sites/emmawoollacott/2017/09/09/exclusiveamazons-fake-review-problem-is-now-worse-than-ever

2017). They assume a fraudulent review may contain inconsistent statistical information deviating from historical reviewing activities. In other words, they identify fraudulent reviews by abnormal reviewing statistical information. For example, these methods may classify the reviews posted by an inactive user within a short period as fraud. Also, they assume fraudulent reviews may be posted by a group of fraudsters who work together to manipulates the opinion of products by posting fraudulent reviews. For example, 100 fraudsters may work together to manipulate the opinion of a product by post all good reviews to this product when it just published in the social media. This co-working will generate abnormal social relations (e.g., social link with a very high density). Thus, they detect fraudulent sentiment according to abnormal social relations. The existing methods have shown remarkable performance in fraudulent sentiment detection because both the abnormal statistical information and abnormal social relation have a good distinguishing ability between honest and fraudulent reviews (Rayana & Akoglu 2016).

However, most of the existing fraudulent sentiment analysis methods may face a cold-start problem when meeting a review that is posted by a new user who has never posted reviews before. When facing the cold-start problem, the existing methods do not have sufficient statistical information or social relation to detect fraudulent sentiment. The cold-start problem is caused by the following reasons. (i) A new user does not have historical information for statistical analysis (You et al. 2018), which is required by most of the existing fraudulent sentiment analysis methods (Ye & Akoglu 2015, Rayana & Akoglu 2015). (ii) A new user does not show any observed social relation, invalidating to detect the potential fraudulent reviewing activities of groups of fraudsters (Liu et al. 2017, Hooi et al. 2016).

The representative methods that can solve the cold-start problem is text-based methods (Mukherjee, Venkataraman, Liu & Glance 2013, Lim et al. 2010, Li et al. 2011). These methods identify a cold-start review (i.e., a review posted by a new user) by only considering patterns in review text, such as abnormal review length and a large proportion of capital words. Thus, these methods avoid the adverse effects brought by lacking historical reviewing activities in the cold-start problem. Recent efforts further embed the co-occurred relations between users, items, and reviews into a vector representation of review text, resulting in significantly better detection performance (You et al. 2018, Wang, Liu & Zhao 2017).

However, an “indistinguishable problem" may arise in cold-start fraudulent sentiment analysis : text-based methods may fail to distinguish fraudulent reviews from honest ones when these reviews have the same text. For example, text-based methods cannot identify whether a review “the product is good" is fraud or honest, because a fraudster can imitate this review for a terrible product (Hooi et al. 2016). As a result, text-based methods are ineffective when dealing with fraudulent sentiment analysis in real-life social media (Mukherjee, Venkataraman, Liu & Glance 2013).

This chapter tackles the cold-start problem by establishing an inferable representation space through learning a co-occurrence-based user reviewing behavior. This reviewing behavior is reflected by the co-occurrence of four review elements: user, item, review text, and review rating. Given a review, regardless of a cold-start review or non-cold-start review, the representation of one element can be inferred from the other elements according to their available co-occurrence-based user reviewing behavior. Based on these inferable space, this chapter further embeds the statistical information and social relations for fraudulent sentiment analysis. Thus, possible fraud can be detected through this inferable space.

This chapter proposes two novel inferable representation learning methods: **j**oint b**e**havior and **s**ocial rela**t**ion inf**er**able embedding (JESTER) in the supervised case and **u**nsupervised **r**eviewing **be**havior **r**epresentation learning (URBER) in the unsupervised case. The proposed methods first embed the user reviewing behavior into a representation space and then transform this representation space in a fraud-sensitive representation space (i.e., a representation space that contains some explicit hints, such as fraud labels) to identify fraudulent sentiment. These two methods share the same user reviewing behavior embedding procedure to build the inferable representation but have different fraud-sensitive representation transformation procedures regarding supervised and unsupervised scenarios. When facing the cold-start problem, they can infer a new user's representation from the representations of the co-occurred review item, review text, and review rating through a closed-form solution. This inferred representation reflects the most probable statistical information and social relations of the new user. With the estimated statistical information and social relations, the proposed methods enable an effective cold-start fraudulent sentiment analysis, which solves the indistinguishable problem in the existing text-based cold-start fraudulent sentiment analysis methods.

This chapter delivers the following significant contributions to fraudulent sentiment analysis:

- This chapter proposes two representation learning methods for supervised and unsupervised cold-start fraudulent sentiment analysis, respectively. The proposed method effectively infers the representation of a new user to estimate the new user's most probable statistical information and social relations, which solves the indistinguishable problem in cold-start fraudulent sentiment analysis.

- This chapter proposes a novel user reviewing behavior embedding method. This method embeds the defined co-occurrence-based user reviewing behavior into a vector space. The embedded user reviewing behavior enables an efficient closed-form solution for the inferring of a new user representation.
- This chapter leverages fraud-sensitive information from social relations to enhance the fraud detection performance of the representation learned from user reviewing behavior. The leveraged information provides more evidence for fraudulent sentiment analysis, especially for detecting reviews manipulated by a group of fraudsters.

This chapter conducts comprehensive experiments on four large real-world data sets. The experimental results demonstrate the effectiveness of the proposed models compared with three state-of-the-art and two baseline methods.

The rest of this chapter is organized as follows. Section 6.2 introduces the proposed inferable representation learning framework. Sections 6.4 and 6.6 give the details of the proposed supervised and unsupervised fraudulent sentiment analysis methods, respectively. Section 6.8 evaluates the proposed fraudulent sentiment analysis methods in terms of different metrics. Lastly, Section 6.9 concludes this chapter and discusses prospects of future research.

6.2 The Framework of Inferable Representation Learning for Fraudulent Sentiment Analysis

6.2.1 Preliminaries

A social reviewing data set S contains a set of users $U = \{u_1, u_2, \cdots, u_{n_u}\}$, a set of items $T = \{t_1, t_2, \cdots, t_{n_t}\}$, a set of review text $D = \{d_1, d_2 \cdots, d_{n_d}\}$, and a set of review rating $R = \{r_1, r_2, \cdots, r_{n_r}\}$. In a social reviewing data set, each user or item is represented as an unique ID, and each review rating is represented as a discrete value (e.g., “high”, “medium”, “low”; “1”, “2”, “3”, “4”, “5”). In other words, user, item, and review rating belong to categorical data. On the contrary, review text belongs to textual data.

A social reviewing data set S can be modeled as a bipartite graph $G = (U, T, E)$, where U and T are as the vertices on two sides of G, respectively, and $E = \{< u, t, d, r >$

$| < u,t,d,r > \in S\}$ defines the edges. Here, each edge $e \in E$ represents a reviewing activity $< u,t,d,r >$, which is the co-occurrence of four elements, including a user u, an item d, a review text t, and a review rating r (i.e., a user u writes a review text t for an item d with a rating r). And the edge e_{u_i,t_j} in E carries a non-negative weight w_{u_i,t_j}, reflecting the social relation strength between user u_i and an item t_j, and the weight w_{u_i,t_j} will be one if the user u_i reviewed the item t_j and be zero if the user u_i does not review the item t_j. Accordingly, the weights in the bipartite graph can be represented by a $n_u \times n_t$ matrix $\mathbf{W} = [w_{u_i,t_j}]$.

6.2.2 Framework

The framework of the proposed inferable representation learning method for cold-start fraudulent sentiment analysis is shown in Figure 6.1. Given a reviewing activity $< u,d,t,r >$, this framework adopts a representation learning module to represent u, t, d, r to m-dimensional vector representations $\mathbf{u}$, $\mathbf{t}$, $\mathbf{d}$, $\mathbf{r} \in \mathscr{R}^m$, respectively. These representations are then fed into a downstream fraud detection module (such as a classifier in supervised cases and a clustering method in unsupervised cases) for fraudulent review detection. The representation learning module of this framework consists of two parts: inferable representation space building and fraud-sensitive information embedding. The inferable representation space building part constructs an inferable space where the representation of one review element can be inferred from the other review elements' representations. The fraud-sensitive information embedding part embeds the information that can be used to identify fraudulent reviews into the elements' representation. This embedded fraud-sensitive information provides essential evidence for fraudulent review detection. By jointly building inferable representation and embedding fraud-sensitive information, the representation learning module can estimate the most probable evidence of a new user for cold-start fraudulent sentiment analysis.

The proposed framework representation can be implemented to different methods. This chapter proposes two methods to implement this framework. These two methods share the same inferable representation learning part to tackle the cold-start problem but have different fraud-sensitive information embedding parts and fraud detection modules catering for supervised and unsupervised cases, respectively. The rest of this chapter first proposes the inferable representation space building method and introduces the new user's representation inferring method in this space. Then, two fraud-sensitive information embedding methods as well as their corresponding fraud detection methods are proposed.

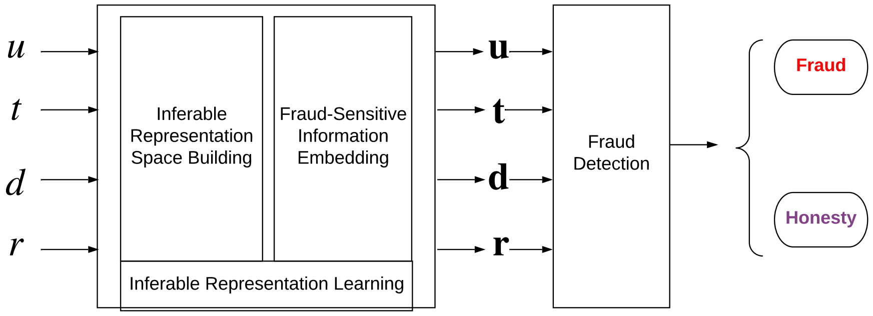

u
t
d
r
Inferable Representation Space Building
Fraud-Sensitive Information Embedding
Inferable Representation Learning
u
t
d
r
Fraud Detection
Fraud
Honesty

Figure 6.1: The Framework of Inferable Representation Learning for Fraudulent Sentiment Analysis

6.3 Inferable Representation Space Building

To build the inferable representation space, a representation learning network is needed to transform review elements to vector representations, and one element's representation can be inferred from the other elements' representations. The architecture of the proposed inferable representation space building method is shown in Figure 6.2. In the following parts, Section 6.3.1 introduces the representation networks, including user embedding network, item embedding network, rating embedding network, and text embedding network. Section 6.3.2 proposes the behavior learning objective to build the inferable representation space. Finally, Section 6.3.3 discusses how to infer a new user's representation after building the inferable representation space.

6.3.1 Representation Learning Networks

The networks consist of four parts: user embedding network, item embedding network, text embedding network, and rating embedding network. The embedding network have a two-layer structure where the first layer is a fully-connected layer with m nodes and the second layer is a normalization layer. While the fully-connected layer maps the one-hot embedding of input (i.e., user ID, item ID, or categorical rating value) to a vector, the normalization layer normalizes the vector to its unit vector. The text embedding network is adopted as the convolutional neural network (CNN) used in (Wang, Liu & Zhao 2017).

6.3.2 Co-occurrence-based User Reviewing Behavior Learning

Inspired by (Wang et al. 2018), this thesis builds the inferable representation space by embedding the co-occurrence-based user reviewing behavior defined below.

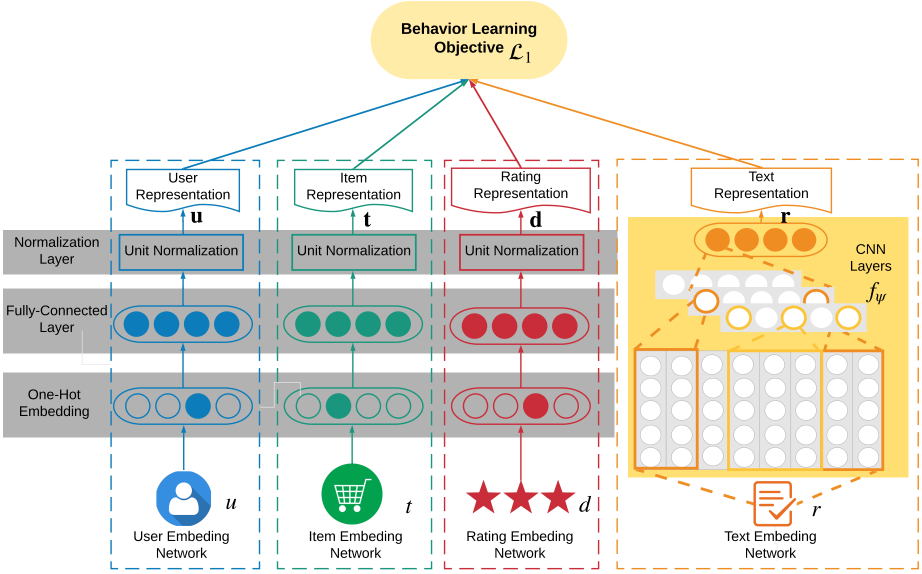


Figure 6.2: Architecture of the proposed Inferable Representation Space Building Method

Definition 6.1. (User Reviewing Behavior) User reviewing behavior is a co-occurrence pattern of the review elements (u, t, d, and r) in user's reviewing activities.

User reviewing behavior is displayed by user historical reviewing activities. Given a social media data set S that consists of a set of reviewing activity, a reviewing activity $< u,t,d,r >$ follows a user reviewing behavior if this reviewing activity is in the data set S (i.e., $< u,t,d,r > \in S$). In other words, a reviewing activity is behavior succeed, if the user, item, review text, and review rating co-occur in history. Otherwise, if u, t, d, and r never co-occurred in history, the reviewing activity $< u,t,d,r >$ does not follow a user reviewing behavior or the reviewing activity is behavior failed. For example, if a user “Tony” only reviews an item “iPhone X” with review text “Awesome phone” and rating “5”, the reviewing activity <Tony, iPhone X, Awesome phone, 5> is behavior succeed, and <Tony, iPhone X, Awesome phone, 3> or <Tony, Dell computer, Awesome phone, 5> is behavior failed. By embedding this co-occurrence-based user reviewing behavior, the absent information of a new user in the cold-start problem could be inferred by from item, review text and review rating according to their co-occurrence relation.

This thesis here introduces a measure to estimate the behavior success rate of a reviewing activity $< u,t,d,r >$. Following (Wang et al. 2018), the behavior displayed by a reviewing activity is first represented as a sum of the vector representations of user, item, review, and rating as follows,

$$\mathbf{b} = \mathbf{u} + \mathbf{t} + \mathbf{d} + \mathbf{r}, \tag{6.1}$$

where $\mathbf{d} = f_\psi(d)$ is a review embedding calculated by a neural network f_ψ with the parameters ψ. Then, the norm of vector $\mathbf{b}$ is used to measure the behavior success rate of the reviewing activity. Specifically, a vector $\mathbf{b}$ with larger norm implies a larger behavior success rate. Consequently, as shown in Figure 6.3, a reviewing activity $< u,t,d,r >$ has a higher behavior success rate if the vector orientation of $\mathbf{u},\mathbf{t},\mathbf{d},\mathbf{r}$ are more similar in the representation space. Considering a reviewing activity either exists in history or not, a reviewing activity is either behavior succeed or failed for a given social media data set. Accordingly, the behavior success rate is mapped to a probability close to 1 or 0 by the following behavior success probability function,

$$s(< u,t,d,r >) = 2 \cdot \frac{1}{1+e^{-\|\mathbf{b}\|_2}} - 1. \tag{6.2}$$

This thesis denotes the observed behavior success probability as $\hat{s}(\cdot)$, where $\hat{s}(< u,t,d,r >$ $) = 1$ if $< u,t,d,r >$ co-occurred in the given social media data set, and $\hat{s}(< u,t,d,r >) = 0$ otherwise. While $s(\cdot)$ describes the behavior success distribution in the representation space, $\hat{s}(\cdot)$ reflects the observed behavior success distribution in the social media. To embed the user reviewing behavior into the element representations, the proposed method minimizes the KL-divergence between the behavior success distribution in the representation space and the observed behavior success distribution in the social media

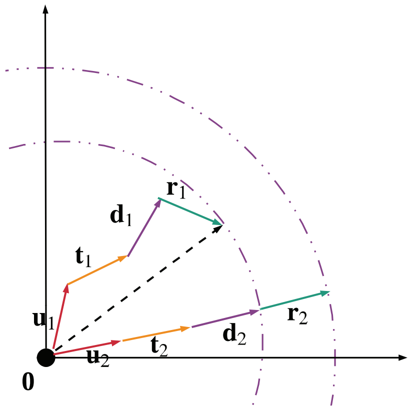


Figure 6.3: The Inferable Representation Space. In this figure, $\mathbf{u},\mathbf{t},\mathbf{d},\mathbf{r}$ refer to the representations of user, item, review, and rating, respectively.

by the following objective function,

$$\min_{\mathbf{u},\mathbf{t},\psi,\mathbf{r}} \sum_{<u,t,d,r>\in S} \hat{s}(<u,t,d,r>)\log\frac{\hat{s}(<u,t,d,r>)}{s(<u,t,d,r>)}. \tag{6.3}$$

Because $\hat{s}(<u,t,d,r>) = 1$ if $<u,t,d,r> \in S$, the Equation (6.3) equals to the follows,

$$\min_{\mathbf{u},\mathbf{t},\psi,\mathbf{r}} -\sum_{<u,t,d,r>\in S} \log s(<u,t,d,r>). \tag{6.4}$$

However, Equation (6.4) only captures the distribution of reviewing activities that follow user reviewing behavior (i.e., $<u,t,d,r> \in S$) but ignores the reviewing activities that do not follow user reviewing behavior (i.e., $<u,t,d,r> \notin S$). In practice, it is impossible to enumerate all reviewing activities because the combination of u, t, d, and r constitutes a huge behavior space. Inspired by the negative sampling used in word2vec (Mikolov et al. 2013), the proposed method randomly samples a set of reviewing activities that do not follow user reviewing behavior (denoted as S_-) and measure the probability of a review activity that does not follow user reviewing behavior in the representation space as follows,

$$s^*(<u,t,d,r>) = 2\cdot\frac{1}{1+e^{\|\mathbf{b}\|_2}}. \tag{6.5}$$

In Equation (6.5), $s^*(<u,t,d,r>)$ will be large (i.e., the review activity $<u,t,d,r>$ is likely to violate user reviewing behavior) if the vector orientations of **u**, **t**, **d**, and **r** are diverse in the representation space. To capture the distribution of reviewing activities that do not follow user reviewing behavior, the proposed method minimizes the KL-divergence between the distribution of behavior failed reviewing activities in the representation space and the sampled behavior failed reviewing activities as follows,

$$\min_{\mathbf{u},\mathbf{t},\psi,\mathbf{r}} -\sum_{<u,t,d,r>\in S_-} \log s^*(<u,t,d,r>). \tag{6.6}$$

Accordingly, the user reviewing behavior learning objective is defined as follows,

$$\mathscr{L}_1 = -\sum_{<u,t,d,r>\in S} \log s(<u,t,d,r>) - \sum_{<u,t,d,r>\in S_-} \log s^*(<u,t,d,r>). \tag{6.7}$$

By minimizing this objective, the co-occurrence-based user reviewing behavior can be embedded into the element representation to build the inferable representation space.

6.3.3 User Representation Inferring in Cold-Start Problem

When facing the cold-start problem, a new user can not be represented by the user representation network because the new user's ID has never been trained. In this case, the proposed method infers the new user's representation in the inferable representation space according to its co-occurrence relation with the item, review text, and review rating, as shown in Figure 6.4. Firstly, the item, review text, and review rating that in the new user's reviewing activity are represented to their representations **t**, **d**, and **r**. Secondly, the user representation is inferred by optimizing the following objective function:

$$\max_{\mathbf{u}} 2 \cdot \frac{1}{1+e^{-\|\mathbf{u}+\mathbf{t}+\mathbf{d}+\mathbf{r}\|_2}} - 1, \tag{6.8}$$

which aims to maximize the behavior success rate (Equation (6.2)) according to the item, review and rating representations.

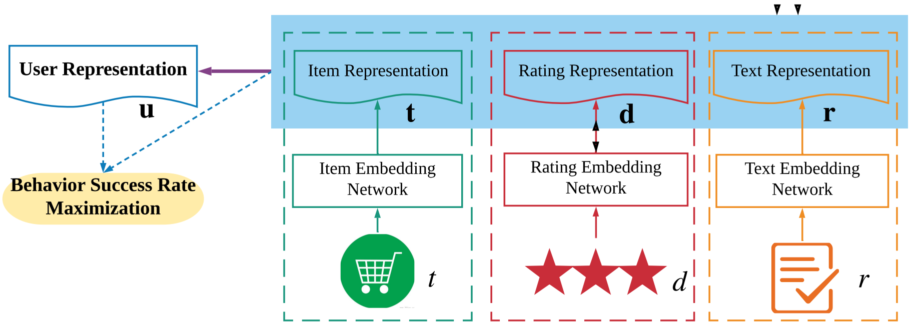


Figure 6.4: User Representation Inferring Process in Cold-Start Problem

Maximizing Equation (6.8) equals to maximizing the $\|\mathbf{u}+\mathbf{t}+\mathbf{d}+\mathbf{r}\|_2$. Given $\mathbf{t}, \mathbf{d}$, and $\mathbf{r}$, as shown in Figure 6.5, the $\mathbf{u}$ that can maximize $\|\mathbf{u}+\mathbf{t}+\mathbf{d}+\mathbf{r}\|_2$ must direct to the direction of $\mathbf{t}+\mathbf{d}+\mathbf{r}$. Because $\mathbf{u}$ is a unit vector, $\mathbf{u}$ equals to the normalization of $\mathbf{t}+\mathbf{d}+\mathbf{r}$ that can be calculated as follows,

$$\mathbf{u} = \frac{\mathbf{t}+\mathbf{d}+\mathbf{r}}{\|\mathbf{t}+\mathbf{d}+\mathbf{r}\|_2}. \tag{6.9}$$

Accordingly, the representation of a new user can be inferred efficiently by the closed-form solution, Equation 6.9.

6.4 Supervised Fraud-Sensitive Information Embedding

In the supervised case, this thesis considers both social relation and statistical information leveraged by fraud labels as fraud-sensitive information. While the social relation

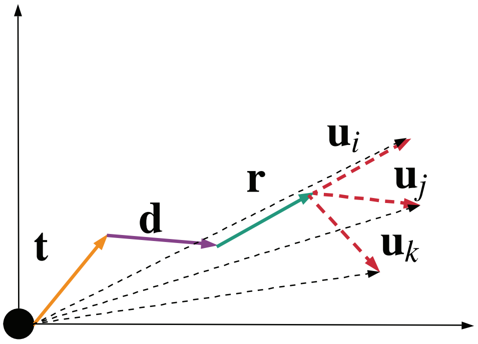


Figure 6.5: The New User Representation Inferring.

reflects the possible fraud reviews manipulated by a group of fraudsters, the statistical information provides fraud evidence directly related to fraud labels. This thesis proposes the JESTER method, which jointly embeds the user reviewing behavior (in Section 6.3.2), social relation (in Section 6.4.1), and statistical information (in Section 6.4.2) for fraud review detection, as shown in Figure 6.6. The JESTER method wraps the statistical information embedding into an end-to-end learning process of fraud review detection. By optimizing a fraud detection objective, the fraud-sensitive statistical information will be discovered and embedded by the representation learning network. Accordingly, JESTER first represents the elements of a reviewing activity to their vector representations. It then feeds these representations into a neural network for fraud review detection.

In the representation learning process, JESTER simultaneously considers three tasks: *user reviewing behavior learning*, *social relation preservation*, and *fraud review detection*, corresponding to three jointly optimized learning objective functions: *behavior learning objective*, *social relation preservation objective*, and *fraud detection objective*. The intuition is that user reviewing behavior learning establishes the inferable ability, the social relation preservation captures the social relations, and the fraud review detection leverages the fraud-related statistical information. By jointly optimizing these three loss functions, JESTER learns elements' inferable representations for fraud review detection.

6.4.1 Social Relation Embedding

The JESTER method first discoveries social relations and then embeds them into elements' representations, as illustrated in Figure 6.7. Specifically, JESTER discoveries both explicit and implicit social relations on the bipartite graph G of a social media data set S. Here, explicit relation refers to the relation directly shown by an edge between a

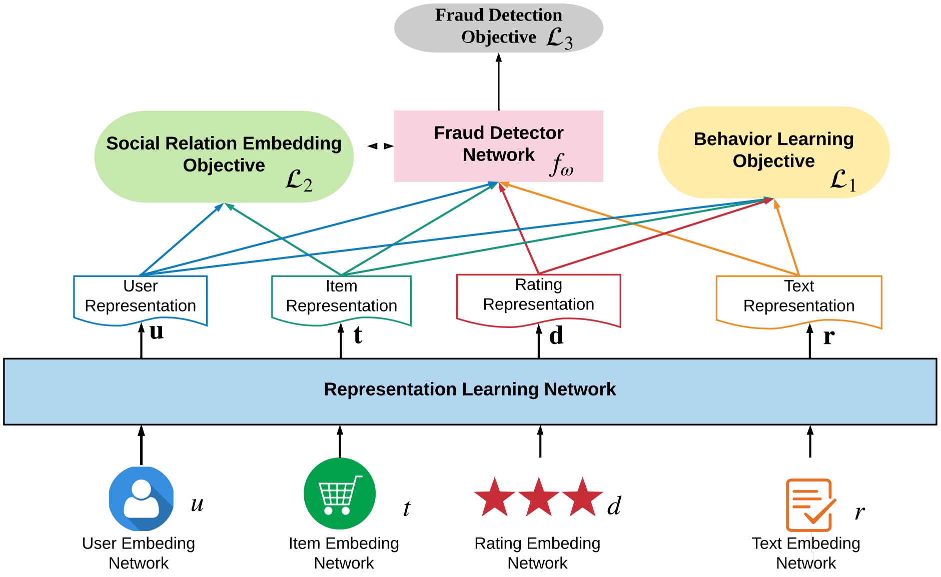

Fraud Detection Objective $\mathcal{L}_3$
Social Relation Embedding Objective $\mathcal{L}_2$
Fraud Detector Network f_ω
Behavior Learning Objective $\mathcal{L}_1$
User Representation
Item Representation
Rating Representation
Text Representation
u
t
d
r
Representation Learning Network
u
User Embeding Network
t
Item Embeding Network
d
Rating Embeding Network
r
Text Embeding Network

Figure 6.6: Architecture of the Proposed JESTER Method

user and an item, which reflects user-item relation; implicit relations refer to user-user relation and item-item relation that is not directly shown by an edge but are hidden in a path consists of a sequence of edges with shared vertices.

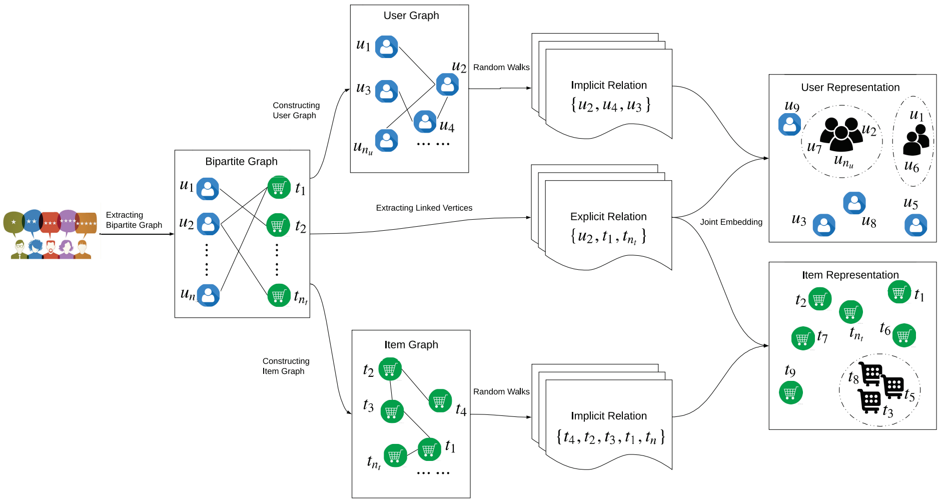


Figure 6.7: The Users/Items Social Relation Embedding Workflow.

Explicit Relations Embedding Explicit relation reflects a user's preference for items. To embed this preference, JESTER assumes a user's representations and an item's representation should be similar if the user prefers the item. In this way, the preference of a user to an item can be measured by the vector similarity in the representation space. Considering user representation $\mathbf{u}$ and item representation $\mathbf{t}$ are all unit vectors, their similarity can be measured by the norm of their sum, $\|\mathbf{u}+\mathbf{t}\|$. The larger the norm is, the more similar $\mathbf{u}$ and $\mathbf{t}$ are. Accordingly, the explicit relation embedding process maximizes $\|\mathbf{u}+\mathbf{t}\|$ if u and t co-occurred in a reviewing activity and minimizes $\|\mathbf{u}+\mathbf{t}\|$ if u and t never co-occurred.

Following the above embedding objective, the explicit relations embedding has already been wrapped in the user reviewing behavior learning objective in Equation (6.7). Specifically, if a user u and an item t co-occurred in a reviewing activity (i.e., $< u,t,d,r >\in S$), the reviewing activity is behavior succeed. By optimizing the loss function Equation (6.7), $\mathbf{u}$ and $\mathbf{t}$ will have similar vector orientations in the representation space and $\|\mathbf{u}+\mathbf{t}\|$ will be large, as shown in Figure 6.3. Therefore, explicit relation has been embedded in the representation space.

Implicit Relations Embedding Implicit relation reveals the potential similarity between users and items. Similar to (Gao et al. 2018), JESTER reconstructs the bipartite graph G into two graphs $G^{(u)}$ and $G^{(t)}$ to discover the implicit relations, where $G^{(u)}$ contains only user vertices U, and $G^{(t)}$ contains only item vertices T. In $G^{(u)}$, u_i and u_j will have an edge e_{u_i,u_j} if existing an item vertex t_k that $e_{u_i,t_k}\in E$ and $e_{u_j,t_k}\in E$, where E is the edge set of G. In $G^{(t)}$, t_i and t_j will have an edge e_{t_i,t_j} if existing a user vertex u_k that $e_{u_k,t_i}\in E$ and $e_{u_k,t_j}\in E$, where E is the edge set of G. Similar to (Deng et al. 2009), JESTER calculates the weights of e_{u_i,u_j} and e_{t_i,t_j} as follows,

$$w_{u_i,u_j}=\sum_{e_{u_i,t_k},e_{u_j,t_k}\in E} w_{u_i,t_k}\cdot w_{u_j,t_k}, \tag{6.10}$$

or

$$w_{t_i,t_j}=\sum_{e_{u_k,t_i},e_{u_k,t_j}\in E} w_{u_k,t_i}\cdot w_{u_k,t_j}. \tag{6.11}$$

To embed the implicit relation, JESTER needs to discover the paths in the graph $G^{(u)}$ and $G^{(t)}$. However, counting all paths in $G^{(u)}$ and $G^{(t)}$ has a great high complexity, which is impracticable for social media data. Inspired by DeepWalk (Perozzi et al. 2014), JESTER performs a truncated random walks on a graph from each node, where the weight of an edge is proportional to the walking probability on the edge. Subsequently, JESTER adopts the walked edges as the paths to reveal implicit relations. In other words, two vertices are treated having an implicit relation if they are in a random walk path. The path generation procedure generates a set of random walk paths $W^{(u)}$ of U and a set of random walk paths $W^{(t)}$ of T.

The JESTER method assumes two users or items always on the same path have an implicit relation and they will have or be affected by a similar user reviewing behavior. It has this assumption because the users or items that always on the same path either have a similar preference (or characteristics) or have a collaboration, which may cause their similar user reviewing behavior. To seamlessly integrate implicit relations with user reviewing behavior, JESTER maximizes the similarity of the orientations of vector of users and items in the representation space if they are in the same path. In this way, these users and items will have similar user reviewing behavior success probability according to Equation (6.2). The similarity of the orientations of vectors can be measured by the cosine similarity. Since all representations are unit vector, the cosine similarity can be reduced to an inner dot of two vectors. Accordingly, JESTER can calculate the probability of two vertices (can either be user or item) in a path from their vector representations as follows,

$$p(v_i, v_j) = 2 \cdot \frac{1}{1 + e^{-\mathbf{v}_i^\top \mathbf{v}_j}} - 1, \tag{6.12}$$

where $\mathbf{v}$ is the vector representation of v. For the vertices that are not in a path, JESTER calculates their probability as

$$p^*(v_i, v_j) = 2 \cdot \frac{1}{1 + e^{\mathbf{v}_i^\top \mathbf{v}_j}}. \tag{6.13}$$

Similar to Equation (6.7), JESTER minimizes the KL-divergence between the distribution of vertices in a path in the representation space and the observed distribution of vertices in a path in social media data. Consequently, JESTER formalizes the social relation embedding objective function as follows,

$$\begin{aligned}
\mathscr{L}_2 = & - \sum_{u_i \in P \wedge P \in W^{(u)}} \sum_{u_j \in C_P(u_i)} \log p(u_i, u_j) \\
& - \sum_{u_i \in P \wedge P \in W^{(u)}} \sum_{u_j \in C_-(u_i)} \log p(u_i, u_j) \\
& - \sum_{t_i \in P \wedge P \in W^{(t)}} \sum_{t_j \in C_P(t_i)} \log p^*(t_i, t_j) \\
& - \sum_{t_i \in P \wedge P \in W^{(t)}} \sum_{t_j \in C_-(t_i)} \log p^*(t_i, t_j),
\end{aligned} \tag{6.14}$$

where P refers to a path in $W^{(\cdot)}$, $C_P(\cdot_i)$ refers to the other vertices of in the path P instead of u_i or t_i, and $C_-(\cdot_i)$ refers to the negative sampled vertices that do not in any path that contains u_i or t_i.

6.4.2 Statistical Information Embedding

The JESTER method leverages statistical information from review elements through an end-to-end learning process supervised by annotated fraud label. Specifically, JESTER fed the elements' representations to a fraud detector network to predict fraud labels. It jointly trains the representation learning network and fraud detector network to discover and embed fraud-sensitive statistical information into elements' representations.

Denoting the fraud detector network in JESTER as f_ω, the predicted fraud label l for a reviewing activity $< u,t,,d,r >$ is obtained by:

$$l(< u,t,,d,r >) = f_\omega(\mathbf{u},\mathbf{t},\mathbf{d},\mathbf{r}), \tag{6.15}$$

where ω refers to the parameters of the fraud detector network. In the learning process, JESTER adopts cross-entropy to evaluate the loss of the fraud detector network. Denoting the supervised fraud review label of $< u,t,,d,r >$ as $\hat{l}(< u,t,,d,r >)$, the objective function of the fraud detector network can be formalized as follows,

$$\begin{aligned} \mathscr{L}_3 = \sum_{<u,t,,d,r>\in S} & -\hat{l}(b)\log l(< u,t,,d,r >) \\ & -(1-\hat{l}(< u,t,,d,r >))\log(1-l(< u,t,,d,r >)). \end{aligned} \tag{6.16}$$

The JESTER method jointly optimizes the user reviewing behavior learning objective, the social relation embedding objective, and the fraud detector network objective to learn the inferable elements' representations. The joint objective function is as follows,

$$\mathscr{L}_J = \alpha_1\mathscr{L}_1 + \alpha_2\mathscr{L}_2 + \alpha_3\mathscr{L}_3 \tag{6.17}$$

where α_1, α_2, and α_3 are hyper-parameters that control the affects of three objectives: $\mathscr{L}_1$, $\mathscr{L}_2$, and $\mathscr{L}_3$.

6.5 Supervised Fraud Detection Method

The JESTER detects fraud reviews by a neural network f_ω based on the inferable elements' representation. In this thesis, f_ω is implemented by a fully-connected neural network with the concatenate of elements' representation vectors as the input and the fraud label as the output. In the fully-connected neural network, JESTER uses the rectified linear unit ($ReLU$) as the activation function of all hidden layers and use the $sigmoid$ as the activation function in the output layer. The number of hidden layers and the number of nodes in each hidden layer are two hyper-parameters that can be adjusted according to different data. As discussed in Section 6.4.2, f_ω is jointly learned with the representation network in an end-to-end learning process.

Given a reviewing activity $< u, t, d, r >$, in the detecting process, JESTER first represents the review elements to their representation $\mathbf{u}$, $\mathbf{t}$, $\mathbf{d}$, and $\mathbf{r}$ through the learned representation learning network. It should be noted that the representation learning network cannot directly generate the representation of a new user when facing the cold-start problem because the representation learning network never trains on the new user's unique ID. In this case, JESTER infers $\mathbf{u}$ from $\mathbf{t}$, $\mathbf{d}$, and $\mathbf{r}$ according to Equation (6.9). After that, JESTER fed these elements' representations into the learned fraud detector network f_ω to predict the fraud label of the reviewing activity.

6.6 Unsupervised Fraud-Sensitive Information Embedding

In the unsupervised case, this thesis discovers fraud-sensitive information only from social relations. Accordingly, this thesis proposes the URBER method for unsupervised cold-start fraud review detection. The URBER method mines fraud-sensitive information from social relations to learn fraud-sensitive representation, which tackles the label-absent problem in unsupervised scenarios. The principle is that abnormal social relations can be used to precisely detect fraud reviews by the existing fraud review detection methods (e.g., dense sub-graph mining) (Hooi et al. 2016); the mined fraud-sensitive information can then be integrated with the embedded reviewing behavior to form fraud-sensitive inferable element representations.

Accordingly, URBER has three principal components as shown in Figure 6.8: *inferable representation space building*, *social relation mining*, and *fraud-sensitive information embedding*. The inferable representation space building component is already introduced in Section 6.3. The social relation mining component adopts a dense sub-graph mining method to generate pseudo fraud labels. The pseudo fraud labels then are used to transform the inferable element representation to fraud-sensitive inferable element representation in the fraud-sensitive information embedding component.

In the transformation process, URBER preserves the information contained in the original inputs (i.e., user, item, review, and rating) by maximizing the mutual information between the original inputs and the transformed fraud-sensitive representations. After the transformation process, URBER further enhances the dense graph mining by adjusting the graph weight according to the density in the transformed representation

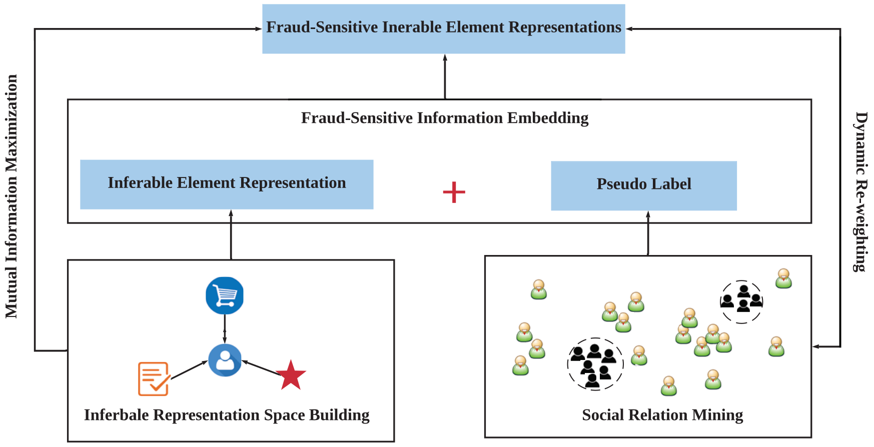

Fraud-Sensitive Inerable Element Representations
Mutual Information Maximization
Dynamic Re-weighting
Fraud-Sensitive Information Embedding
Inferable Element Representation
+
Pseudo Label
Inferbale Representation Space Building
Social Relation Mining

Figure 6.8: Architecture of the Proposed URBER Method

space. It then repeats these processes to re-generate the pseudo-labels and re-transform the element representations iteratively.

6.6.1 Social Relation Mining

Motivated by (Liu et al. 2017, Hooi et al. 2016), URBER leverages social relations to integrate fraud-sensitive information to inferable representation. This social relation mining process is shown in Algorithm 1. To achieve this goal, URBER greedily removes vertexes in the bipartite graph G of a social media data set S to maximize the sub-graph density per a given density evaluation (Algorithm 1 line 2 - 5). The final remained sub-graph (Algorithm 1 line 6) will be the part that has the largest density in the bipartite graph (as shown in Figure 6.9), thus it can reflect the users who may work together to manipulate reviews.

In Algorithm 1, the density metric $g(\cdot)$ is defined as follows,

$$g(S) = \frac{f(S)}{|S|}, \tag{6.18}$$

where

$$f(S) = \sum_{<u,t,d,r> \in E} w_{u,t}, \tag{6.19}$$

and $w_{u,t} \in \mathscr{R}^+$ refers to the link weight between user u to item t. Initially, URBER assigns all link weights as 1. In the learning process, it adopts a dynamic re-weighting strategy to iteratively update the link weights.

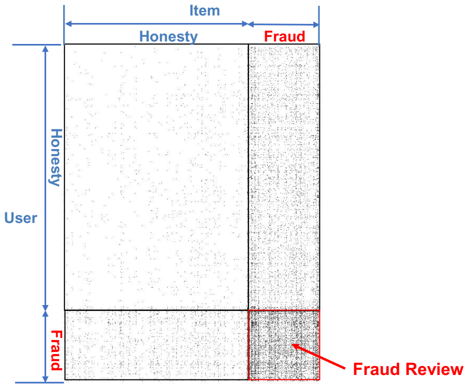


Figure 6.9: Density in a User-Item Bipartite Graph

Algorithm 1 Social Relation Mining of URBER.

```
**Input:** Bipartite graph $G = (U, T, E)$;
**Output:** The pseudo-labels set $Y$;
 1: $X_0 \leftarrow U \cup T$
 2: for $t = 1, \cdots, (n_u + n_t)$ do
 3:     $i^* \leftarrow \arg\max_{i \in X_{t-1}} g(X_{t-1} \backslash \{i\})$
 4:     $X_t \leftarrow X_{t-1} \backslash \{i^*\}$
 5: end for
 6: $X^* \leftarrow \arg\max_{X_i \in \{X_0, \cdots, X_{n_u + n_t}\}} g(X_i)$;
 7: for $u = u_1, \cdots, u_{n_u}$ do
 8:     if $u \in X^*$ then
 9:         $y_i = c_f$
10:     else
11:         $y_i = c_n$
12:     end if
13: end for
14: return $Y = \{y_1, \cdots, y_{n_u}\}$
```

The URBER method generates pseudo-labels for a reviewing activity according to the dense sub-graph mining results. Specifically, it gives a pseudo fraud label (c_f) to each reviewing activity in the detected dense sub-graph (i.e., the user and item of an reviewing activity are both in the dense sub-graph), and assigns pseudo normal labels (c_n) to others (Algorithm 1 line 7 - 14). These pseudo-labels inherit the social relations and will be used in the following fraud-sensitive information embedding component and fraud detection.

6.6.2 Fraud-Sensitive Information Embedding

The URBER method adopts its generated pseudo label to embed fraud-sensitive information into element representation. Specifically, it uses fraud-information embedding network $f_{\mathbf{p}_u}(\cdot)$, $f_{\mathbf{p}_t}(\cdot)$, $f_{\mathbf{p}_d}(\cdot)$, and $f_{\mathbf{p}_r}(\cdot)$ to transform inferable element representations $\mathbf{u}$, $\mathbf{t}$, $\mathbf{d}$, and $\mathbf{r}$ to fraud-sensitive inferable element representation $\mathbf{u}^*$, $\mathbf{t}^*$, $\mathbf{d}^*$, and $\mathbf{r}^*$ by minimizing a pseudo-labels prediction objective. Here, the pseudo-labels prediction objective is defined as the cross-entropy between pseudo-labels and labels predicted by a classifier based on $\mathbf{u}^*$, $\mathbf{t}^*$, $\mathbf{d}^*$, and $\mathbf{r}^*$.

The fraud-sensitive information embedding network is implemented by four fully-connected neural networks, each of which maps an element representation to its fraud-sensitive embedding. In these networks, URBER adopts the *ReLU* as the activation function in each hidden node. The number of hidden layers and the number of nodes in each hidden layer are hyper-parameters that can be adjusted according to different data.

Directly conducting this embedding procedure has a risk of causing an overfitting hazard. The transformed fraud-sensitive inferable element representation may be dominated by the pseudo labels; thus, it may lose original information contained in reviewing activities. To avoid this overfitting hazard, URBER further maximizes the mutual information between the original reviewing activity and the fraud-sensitive inferable element representation.

Accordingly, the objective function of URBER's fraud-sensitive information embedding can be formalized as follows,

$$
\begin{aligned}
\min_{P,\mathbf{w},b} \quad & \sum_{i=1}^{n_s} \sum_{y=\{c_f,c_n\}} \mathbf{1}[y_i = y] \log q_i + I(S^*;S) \\
s.t. \quad & q_i = \text{softmax}(\mathbf{w} \cdot [\mathbf{u}_i^*, \mathbf{t}_i^*, \mathbf{d}_i^*, \mathbf{r}_i^*] + b), \\
& \mathbf{u}_i^* = f_{\mathbf{p}_u}(\mathbf{u}_i), \\
& \mathbf{t}_i^* = f_{\mathbf{p}_t}(\mathbf{t}_i), \\
& \mathbf{d}_i^* = f_{\mathbf{p}_d}(\mathbf{d}_i), \\
& \mathbf{r}_i^* = f_{\mathbf{p}_r}(\mathbf{r}_i),
\end{aligned}
\tag{6.20}
$$

where y_i is the pseudo-label of the i-th user assigned by Algorithm 1, where n_s is the number of the existing reviewing activities, where $P = \{\mathbf{p}_u, \mathbf{p}_t, \mathbf{p}_d, \mathbf{p}_r\}$, $\mathbf{w}$, and b are the parameters of a softmax function, and where $I(\cdot;\cdot)$ is a mutual information measurement. This thesis adopts the mutual information neural estimator (Belghazi et al. 2018) as $I(\cdot;\cdot)$ for the computational convenience.

6.6.3 Dynamic Re-Weighting Strategy

The discriminative ability of the fraud-sensitive representation should be strong, because the representation will be used to detect fraud reviews. In URBER, this discriminative ability is mainly obtained from the pseudo-labels generated by the dense sub-graph mining in the social relation mining component. However, social relation may not comprehensively indicate all kinds of fraud reviews (Rayana & Akoglu 2016). As a result, the discriminative ability of fraud-sensitive representation may not be good if learning only from the social relation.

To enhance the discriminative ability, URBER reinforces the focusing of the dense sub-graph mining on the suspicious users discovered in the fraud-sensitive inferable element representation space, which reflects both the reviewing behavior and the social relation. Specifically, URBER clusters a set of users into two categories according to their fraud-sensitive inferable representations. It then re-weights the link of each user by the reciprocal of the number of its assigned categories. Formally, the link weight of a user u is assigned as,

$$w_{u,\cdot} = \frac{1}{|C_u|}, \tag{6.21}$$

where C_u refers to a set of users with the same category as user u, and $|\cdot|$ returns the size of the set. The assumption behind this re-weighting is that a user with less similar users is more suspicious as a fraudster. After re-weighting the links, URBER conducts the dense sub-graph mining again to generates new pseudo-labels, which are further integrated with the embedded reviewing behavior to form the element fraud-sensitive representation. The URBER method repeats this dynamic re-weighting strategy until convergence. With the dynamic link re-weighting strategy, the fraud-sensitive information embedding procedure of URBER is summarized in Algorithm 2.

6.7 Unsupervised Fraud Detection Method

The URBER method adopts different approaches to detect fraud reviews for the review posted by the existing users and new users, respectively. For a review posted by an existing user, URBER assigns the label by conducting Algorithm 1. For a review posted by a new user, URBER first finds the k-nearest neighbors of the new user in the fraud-sensitive inferable user representation space, and it then assigns the label that appears mostly in the k-nearest neighbors as the predicted label of the new user.

Algorithm 2 Fraud-Sensitive Information Embedding of URBER with Dynamic Re-Weighting Strategy.

```
Input: Online review set $S$, convergence threshold $\epsilon$;
Output: Element fraud-sensitive representation $\{\mathbf{u}^*, \mathbf{t}^*, \mathbf{d}^*, \mathbf{r}^*\}$;
Embedding reviewing behavior by optimizing Equation (6.7)
Generating pseudo-label set $Y$ by Algorithm 1
Generating element fraud-sensitive representation $\{\mathbf{u}^*, \mathbf{t}^*, \mathbf{d}^*, \mathbf{r}^*\}$ by Equation (6.20)
Initializing $\Delta = +\infty$
while $\Delta > \epsilon$ do
    $Y' \leftarrow Y$
    Clustering $\mathbf{u}^*$ into two categories
    Re-weighting user-item graph links by Equation (6.21)
    Generating pseudo-label set $Y$ by Algorithm 1
    Generating element fraud-sensitive representation $\{\mathbf{u}^*, \mathbf{t}^*, \mathbf{d}^*, \mathbf{r}^*\}$ by Equation (6.20)
    $\Delta = 1 - \frac{\sum_{y_i \in Y, y'_i \in Y'} \mathbb{1}[y_i = y'_i]}{|Y|}$
end while
return $\{\mathbf{u}^*, \mathbf{t}^*, \mathbf{d}^*, \mathbf{r}^*\}$
```

6.8 Experiments and Evaluation of Fraudulent Sentiment Analysis

6.8.1 Data Sets

Following the literature (You et al. 2018, Wang, Liu & Zhao 2017) about cold-start fraud detection, the experiments are carried on four real-life data sets, including Yelp-Hotel, Yelp-Restaurant, Yelp-NYC, and Yelp-Zip, which are also commonly used in previous fraud detection researches (Mukherjee, Venkataraman, Liu & Glance 2013, Rayana & Akoglu 2015, Mukherjee, Kumar, Liu, Wang, Hsu, Castellanos & Ghosh 2013).

6.8.1.1 Supervised Fraudulent Sentiment Analysis

In the supervised case, the experiments split the original Yelp-Zip and Yelp-NYC data sets into several subsets according to the time to evaluate the fraudulent sentiment analysis performance stably. The experiments further split each subset into two parts. The first part includes the reviews posted before a time point, while the second part contains the rest reviews. From the second part, the experiments pick up the reviews that are posted by new users for the first time as cold-start reviews. The experiments train each fraud detection method on the first part and evaluate these methods on the second part. Table 6.1 displays the statistics of the data sets for the evaluation of supervised fraudulent sentiment analysis.

Table 6.1: Statistics of Data Sets for Supervised Fraudulent Sentiment Analysis

Name	**Training Data**		**Testing Data**				
	Time Period	**#R**	**Time Period**	**#F**	**#FC**	**#N**	**#NC**
Zip_1	24/10/08 – 24/03/09	10530	25/03/09 – 25/06/10	6267	4848	43744	15952
Zip_2	24/03/09 – 24/08/09	13252	25/08/09 – 25/12/09	1396	1075	10220	3820
NYC_1	24/10/08 – 24/03/09	6780	25/03/09 – 25/06/10	3183	2539	27974	11313
NYC_2	24/03/09 – 24/08/09	8243	25/08/09 – 25/12/09	748	594	6664	2754

In this table, #R refers to the number of reviews; #F and #FC refer to the number of fraudulent reviews and cold-start fraudulent reviews, respectively; and #N and #NC refer to the number of honest reviews and cold-start honest reviews, respectively.

6.8.1.2 Unsupervised Fraudulent Sentiment Analysis

In unsupervised cases, the experiments split original Yelp-Hotel and Yelp-Restaurant data sets into two parts for fraudulent sentiment analysis performance evaluation. The first part includes 90% earliest posted reviews. The users who posted these reviews are treated as existing users. The second part is the 10% latest posted reviews. Similar to the settings in supervised fraudulent sentiment analysis evaluation, the experiments pick up the reviews which wrote by new users for the first time in the second part as cold-start reviews. Furthermore, the experiments use the whole data sets to evaluate the general fraudulent sentiment analysis performance and do the ablation study. The statistics of these data sets are shown in Table 6.4 and Table 6.5 for the unsupervised cold-start fraudulent sentiment analysis and general fraudulent sentiment analysis, respectively.

6.8.2 Evaluation Metrics

The experiments evaluate the fraudulent sentiment analysis performance of each method by three metrics, including *precision*, *recall*, and *F-score*. Here, the precision evaluates the ratio of the number of correct analysis results to the number of all analysis results, recall reflects the ratio of the number of undetected reviews to the number of all reviews that should be detected, and the F-score indicates an average of precision and recall. The experiments use all of them because the fraudulent sentiment analysis is an imbalanced classification problem (i.e., the number of fraudulent reviews are much less than honest reviews; Luca & Zervas 2016) that cannot be considered from only precision or recall perspective. The experiments report these three metrics per ground-truth honest and fraudulent classes to illustrate the performance for different categories, and the experiments further average them to show the overall performance. Higher precision, recall, and F-score indicate better performance.

The experiments follow the work of Rayana & Akoglu (2015) and Wang, Liu & Zhao (2017) to use the results of the Yelp commercial fake review filter as the ground-truth for performance evaluation. Although the reviews (fraudulent reviews) filtered by this filter and the unfiltered reviews (honest reviews) are likely to be the closest to real fraudulent and honest reviews (Mukherjee, Venkataraman, Liu & Glance 2013), they are not absolutely accurate (Li et al. 2014). The inaccuracy exists because it is hard for the commercial filter to have the same psychological state of mind as that of the fraudsters who have real businesses to promote or to demote, especially in the cold-start problem.

6.8.3 Parameters Settings

The experiments use a CNN network to embed reviews following Wang, Liu & Zhao (2017). The CNN network adopts 100 filters with the size of 3×100 on the pre-trained 100-dimensional word embedding by the GloVe algorithm (Pennington et al. 2014) [3]. The experiments embed the user, item, and rating into a 100-dimension representation vector. The experiments implement the fraud detector network and fraud-sensitive information embedding network by a 3-layer fully-connected neural network with 100 nodes in each hidden layer and use $ReLU$ as the activation function for each hidden node. The JESTER and URBER models are trained by the Adam optimization algorithm (Kingma & Ba 2014) with a batch size of 32. The k-means algorithm is selected as the clustering method in URBER, while the Euclidean distance is used as the distance metric to get 5-nearest neighbors of a new user in the user fraud-sensitive representation space. For the parameters in the compared methods, the experiments take the recommended settings reported in their corresponding papers.

[3] The pre-trained word embedding can be downloaded from http://nlp.stanford.edu/data/glove.6B.zip

6.8.4 Effectiveness on Supervised Cold-start Fraudulent Detection

Experimental Settings. This experiment compares JESTER with state-of-the-art method JETB (Wang, Liu & Zhao 2017). The JETB method handles the cold-start problem by capturing relations among entities (user, item, and review) and embedding these relations into review representation vectors. When a new user posts a new review, JETB can represent this new review by its trained network and classify this new review by its classifier according to the new review's representation vector. In the original literature (Wang, Liu & Zhao 2017), JETB uses SVM as the fraud classifier based on the JETB-generated review representation vectors. However, SVM has a large time complexity, $O(n^3)$, where n is the number of training samples. Accordingly, the vanilla JETB does not suit the problem with a large amount of data. To make JETB practicable, this experiment uses a 3-layer fully-connected neural network instead of SVM as the fraud classifier in JETB.

This experiment further compares JESTER with two review-text-based fraudulent sentiment analysis methods, which are also selected as the compared method by Wang, Liu & Zhao (2017), as baseline competitors. These two methods extract features from review text and feed these features into a classifier for fraudulent sentiment analysis. Specifically, the first method (denoted as Bigram) uses the bigram feature. The second method (denoted as Behavior) uses (i) the bigram feature, (ii) the length of review, (iii) the absolute rating diversity of a review compared with other reviews of the same item, and (iv) the similarity of a review to its most similar reviews of the same item under the cosine similarity. This experiment also uses a 3-layer fully-connected neural network as the fraud classifier in these two methods.

Findings - JESTER Significantly Outperforming that State-of-the-art Cold-start Fraudulent Sentiment Analysis Method. Table 6.2 illustrates the cold-start fraudulent sentiment analysis performance of JESTER compared with JETB, Behavior, and Bigram on four subsets of Yelp-Zip and Yelp-NYC data sets with different periods. The JESTER method gains improvement significantly for cold-start fraud review detection (i.e., 0.11, 0.08, 0.13, and 0.10 F-score increase on Zip_1, Zip_2, NYC_1, and NYC_2, respectively). This averaged performance improvement is mainly contributed by the increased recall of the fraudulent sentiment analysis (corresponding recall increase values are 0.11, 0.13, 0.18, and 0.11). As shown in the results, JESTER slightly “decreases" the performance of honest sentiment analysis. This performance decreasing may be caused by some labels that incorrectly annotated by the Yelp commercial filter. The Yelp commercial filter may overlook some cold-start fraud reviews. In other words, some reviews that labeled as honest in these data sets should be fraud. Because JESTER correctly classified these reviews to fraud, the recall of honest review detection seems decreased.

Table 6.2: Supervised Cold-start Fraud Detection Performance of Different Methods

Data Info.		JESTER			JETB			Behavior			Bigram			Improvement		
Name	Category	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Zip_1	Normal	**0.81**	0.90	0.85	0.77	**1.00**	0.87	0.77	0.99	0.87	0.78	0.96	**0.86**	**0.03**	-0.10	-0.02
	Fraud	0.37	**0.21**	**0.27**	0.24	0.00	0.00	**0.54**	0.05	0.09	0.42	0.10	0.16	-0.17	**0.11**	**0.11**
Zip_2	Normal	**0.82**	0.84	0.83	0.78	**1.00**	0.88	0.79	0.99	0.88	0.80	0.92	**0.85**	**0.02**	-0.16	-0.05
	Fraud	0.33	**0.30**	**0.31**	0.45	0.01	0.02	**0.54**	0.06	0.11	0.37	0.17	0.23	-0.21	**0.13**	**0.08**
NYC_1	Normal	**0.84**	0.84	0.84	0.82	1.00	**0.90**	0.82	1.00	**0.90**	0.82	0.96	0.89	**0.02**	-0.16	-0.06
	Fraud	0.26	**0.26**	**0.26**	0.00	0.00	0.00	**0.38**	0.00	0.00	0.31	0.08	0.13	-0.12	**0.18**	**0.13**
NYC_2	Normal	**0.85**	0.90	0.87	0.82	**1.00**	**0.90**	0.82	**1.00**	**0.90**	0.83	0.94	0.88	**0.02**	-0.10	-0.03
	Fraud	**0.31**	**0.23**	**0.27**	0.00	0.00	0.00	0.00	0.00	0.00	0.29	0.12	0.17	**0.02**	**0.11**	**0.10**

Precision (P), Recall (R) and F-score (F) are reported per normal and fraud reviews. The best results are highlighted in bold.

In addition to review text, JESTER further leverages information from users, items, and ratings. This comprehensive information enables JESTER to capture more fraud evidence from multiple views effectively, and JESTER can use this information even when facing the cold-start problem because of the learned inferable representation. As a result, JESTER can achieve significant performance improvement in cold-start fraudulent sentiment analysis.

6.8.5 Effectiveness on Supervised General Fraudulent Sentiment Analysis

Experimental Settings. This experiment compares JESTER with JETB and two state-of-the-art competitors, including FRAUDER (Hooi et al. 2016) and HoloScope (Liu et al. 2017), in detecting *general fraudulent sentiment* (i.e., the sentiment of all the reviews contained in the testing data set). Different from JETB, which is a review-text-based method, FRAUDER and HoloScopre are two social-relation-based fraudulent sentiment analysis methods. Specifically, the FRAUDER method models the social relation as a graph and detects fraudulent sentiments by dense sub-graph mining. The HoloScope method also adopts a graph to model social relations but detects fraudulent sentiments by jointly considering the graph topology and review temporal spikes.

Findings - JESTER Significantly Improving General Fraudulent Sentiment Analysis Performance. Table 6.3 reports the precision, recall, and F-score of JESTER, JETB, FRAUDER, and HoloScope. Overall, JESTER significantly outperforms the competitors in fraud review detection. It improves 0.16, 0.21, 0.20, and 0.20 compared with the best-performing method in terms of F-score on four data sets for fraudulent sentiment analysis.

Table 6.3: Supervised General Fraud Detection Performance of Different Methods

Data Info.		JESTER			JETB			FRAUDER			HoloScope			Improvement		
Name	Category	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Zip_1	Normal	**0.89**	0.92	0.91	0.87	**1.00**	**0.93**	0.87	0.95	0.91	0.86	0.86	0.86	**0.02**	-0.08	-0.02
	Fraud	**0.23**	**0.17**	**0.19**	0.18	0.00	0.01	0.01	0.00	0.00	0.03	0.03	0.03	**0.05**	**0.14**	**0.16**
Zip_2	Normal	**0.90**	0.87	0.88	0.78	**1.00**	0.88	0.88	0.95	**0.91**	0.87	0.88	0.88	**0.02**	-0.13	-0.03
	Fraud	0.22	**0.29**	0.25	**0.45**	0.01	0.02	0.04	0.02	0.02	0.04	0.04	0.04	-0.23	**0.25**	**0.21**
NYC_1	Normal	**0.91**	0.88	0.90	0.90	**1.00**	0.95	0.88	0.86	0.87	0.88	0.86	0.87	**0.01**	-0.12	-0.05
	Fraud	**0.18**	**0.25**	**0.21**	0.00	0.00	0.00	0.01	0.01	0.01	0.01	0.01	0.01	**0.17**	**0.24**	**0.02**
NYC_2	Normal	**0.92**	0.92	0.92	0.90	**1.00**	**0.95**	0.88	0.82	0.85	0.87	0.69	0.77	**0.02**	-0.08	-0.03
	Fraud	**0.24**	**0.22**	**0.23**	0.00	0.00	0.00	0.01	0.02	0.02	0.02	0.06	0.03	**0.22**	**0.16**	**0.20**

Precision (P), Recall (R) and F-score (F) are reported per normal and fraud reviews. The best results are highlighted in bold.

The dramatic performance improvement of JESTER is mainly contributed by jointly embedding user reviewing behavior and social relations of user and item in its element representations. Compared to FRAUDER and HoloScope that capture social relations, JESTER further considers the user-reviewing behavior to detect personalized fraud effectively. Compared to JETB, JESTER seamlessly integrates social relations of user and item to avoid camouflage. Consequently, JESTER obtains recall improvement up to 0.24 compared with the competitors.

6.8.6 Effectiveness on Unsupervised Cold-start Fraudulent Sentiment Analysis

Experimental Settings. This experiment compares UEBER with the state-of-the-art unsupervised cold-start fraud review detection method SUPER-COLD (Li et al. 2019). The SUPER-COLD and URBER methods have similar unsupervised cold-start fraudulent sentiment analysis mechanism. Their differences are that (1) URBER adopts an inferable representation learning when embedding user reviewing behavior, and (2) URBER further introduces mutual information maximization regularization in fraud-sensitive information embedding to avoid the over-fitting problem.

Findings - URBER Outperforming the State-of-the-art Cold-start Fraud Detection Method. Table 6.4 demonstrates the fraudulent sentiment analysis performance of URBER. The URBER method improves the averaged fraudulent sentiment analysis performance (0.02 and 0.03 F-score increase on Yelp-Hotel and Yelp-Restaurant data sets) compared to SUPER-COLD. These results demonstrate that the inferable representation learning and mutual information maximization regularization introduced by URBER are effective.

Table 6.4: Unsupervised Cold-start Fraud Detection of Different Methods

Data Info.				URBER			SUPER-COLD			Improvement		
Name	Category	#Existing	#Cold-start	P	R	F	P	R	F	P	R	F
Hotel	Normal	376,671	60	**0.48**	**0.15**	**0.23**	0.45	**0.15**	**0.23**	**0.03**	0.00	0.00
	Fraud	242,825	122	**0.73**	**0.92**	**0.81**	0.69	0.91	0.78	**0.04**	**0.01**	**0.03**
	Overall	619,496	182	**0.65**	**0.67**	**0.62**	0.61	0.66	0.60	**0.04**	**0.01**	**0.02**
Restaurant	Normal	412,435	1,654	**0.68**	**0.86**	**0.76**	0.64	0.84	0.73	**0.04**	**0.02**	**0.03**
	Fraud	297,188	873	**0.65**	**0.70**	**0.67**	0.62	0.68	0.65	**0.03**	**0.02**	**0.02**
	Overall	709,623	2,527	**0.67**	**0.80**	**0.73**	0.63	0.78	0.70	**0.04**	**0.02**	**0.03**

Precision (P), Recall (R) and F-score (F) are reported per normal and fraud reviews. The best results are highlighted in bold.

6.8.7 Effectiveness on Unsupervised General Fraudulent Sentiment Analysis

Experimental Settings. This experiment compares URBER with three state-of-the-art competitors, including Frauder (Hooi et al. 2016), HoloScope (Liu et al. 2017), and SPEAGLE (Rayana & Akoglu 2016), in detecting general fraudulent sentiment. These three competitors have different but relevant mechanisms compared to URBER.

- *Fixed weighting dense sub-graph mining-based method - FRAUDER* (Hooi et al. 2016). The FRAUDER method detects fraudulent sentiment by dense sub-graph mining. To detect camouflage and hijacked accounts, it adopts a fixed weighting strategy. Different from FRAUDER, the dense sub-graph mining method used in URBER is with a dynamic link weighting strategy to further fuse the element relation with the social relation.

- *Dynamic weighting dense sub-graph-mining-based method - HoloScope* (Liu et al. 2017). The HoloScope method uses graph topology and temporal spikes to detect fraudsters groups and employs a dynamic weighting approach to enable a more accurate fraudulent sentiment analysis. However, the dynamic weighting is only

Table 6.5: Unsupervised General Fraud Detection of Different Methods

Data Info.			URBER			HoloScope			FRAUDER			SPEAGLE			Improvement		
Name	Category	#Review	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Hotel	Normal	420,785	**0.69**	0.95	**0.8**	0.64	0.6	0.62	0.64	**0.98**	0.77	0.53	-	-	**0.05**	-0.03	**0.18**
	Fraud	267,544	**0.82**	0.33	**0.47**	0.42	**0.46**	0.44	**0.82**	0.11	0.31	0.72	-	-	0.00	-0.13	**0.03**
	Overall	888,329	**0.74**	**0.71**	**0.67**	0.55	0.55	0.55	0.71	0.65	0.55	0.60	-	-	**0.03**	**0.06**	**0.12**
Restaurant	Normal	461,190	**0.68**	0.88	**0.77**	0.51	**0.95**	0.66	0.63	**0.95**	0.76	0.42	-	-	**0.05**	-0.07	**0.01**
	Fraud	326,981	0.72	**0.42**	**0.53**	**0.74**	0.12	0.21	**0.74**	0.21	0.33	0.58	-	-	-0.02	**0.21**	**0.20**
	Overall	788,741	**0.70**	**0.69**	**0.67**	0.63	0.52	0.43	0.68	0.64	0.58	0.49	-	-	**0.02**	**0.05**	**0.09**

Precision (P), Recall (R) and F-score (F) are reported per normal and fraud reviews. The best results are highlighted in bold.

conducted once according to the user temporal spikes. In contrast, URBER interactively updates the dynamic weighting along the user behavior embedding process.

- *Metadata and social relation integration-based method - SPEAGLE* (Rayana & Akoglu 2016). The SPEAGLE method proposes a unified framework to utilize metadata and the social relation in a Markov random field for fraudulent sentiment analysis. While SPEAGLE needs fraud labels, URBER is a completely unsupervised method that jointly considers element relation and social relation for user behavior representation.

While FRAUDER and HoloScope directly predict fraudulent sentiment, SPEAGLE gives a probability of a sentiment that may be fake. To make a fair comparison, this experiment reports only the averaged precision of SPEAGLE but ignores its the recall and F-score.

Findings - URBER Significantly Improving General Fraudulent Sentiment Analysis Performance, Especially in terms of Recall. Table 6.5 reports the precision, recall, and F-score of URBER, Frauder, HoloScope, and SPEAGLE. Overall, URBER significantly outperforms its competitors. It improves 0.12 and 0.09 compared with the best-performing method in terms of F-score on two data sets, respectively.

Unlike FRAUDER and HoloScope that ignore the element relation when they perform dense sub-graph mining based on social relation, URBER couples these two independent relations to iteratively refine their performance by the dynamic link weighting. This mechanism enables URBER to avoid camouflage by considering social relations and to effectively detect personalized fraud by considering element relations. As a result, URBER obtains recall improvement up to 0.21 compared with the competitors.

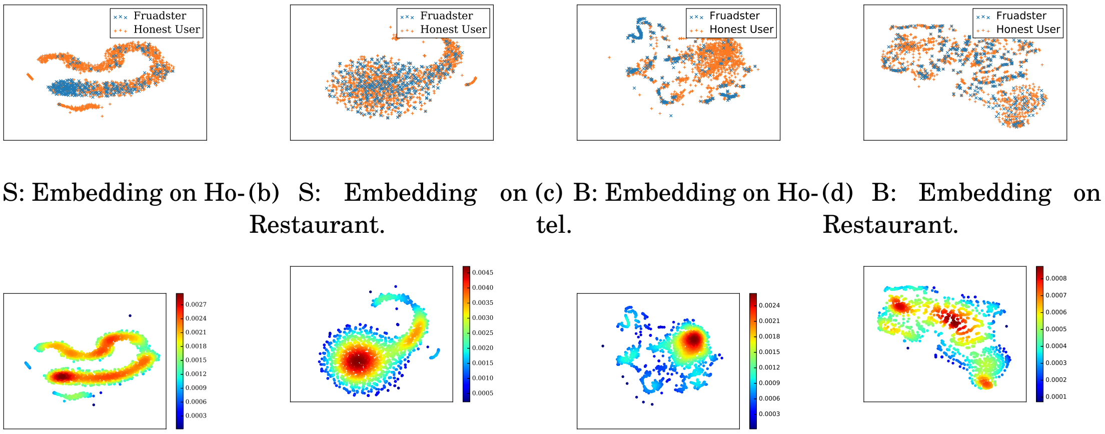
6.8.8 Evaluating the Effectiveness of User Reviewing Behavior and User/Item Social Relations for Fraudulent Sentiment Analysis

Experimental Settings. The experiment visualizes the user representation in a two-dimensional space trough TSNE (Maaten & Hinton 2008), and it plots the ground-truth labels of each user at their positions in the representation space. The user representation learned according to the user reviewing behavior learning loss function, Equation (6.7), is compared with that learned according to the social relation preservation loss function, Equation (6.14), on Yelp-Hotel and Yelp-Restaurant data sets.

Findings - Behavior-embedded Representation contributing to Personalized Fraud Review Detection and Social Relation-embedded Representation Contributes to Collaborative Fraud Review Detection. The behavior-embedded and social relation-embedded user representations are visualized in Figure 6.10. As shown in Figure 6.10, users have more diverse representations in the behavior-embedded representation space compared with social relation-embedded representation space. This diversity indicates that more personalized information is captured by the behavior-embedded representation, which is important to identify personalized fraudulent sentiment. However, in the behavior-embedded representation space, the users with large density are not consistent with the ground-truth fraudster label. In contrast, the density of social relation-embedded representation is consistent with the ground-truth fraudsters distribution. As evidenced by Hooi et al. (2016), the collaborative manipulation of sentiment will generate density connections between users. Accordingly, the results demonstrate that the embedded social relation by the proposed methods is essential for collaborative fraudulent sentiment analysis. A high-quality user representation will enable a dense distribution for fraudsters because of the collaborative manipulation (Hooi et al. 2016). This result qualitatively illustrates that the social relation of users is essential for collaborative fraudulent sentiment analysis.

6.9 Summary

This chapter introduces two novel inferable representation learning methods for fraudulent sentiment analysis with the cold-start problem in supervised and unsupervised cases, respectively. These two methods jointly embed user reviewing behavior and user/item social relations into the inferable representation vectors of users, items, reviews, and



(a) S: Embedding on Hotel.

(b) S: Embedding on Restaurant.

(c) B: Embedding on Hotel.

(d) B: Embedding on Restaurant.

(e) S: Density on Hotel.

(f) S: Density on Restaurant.

(g) B: Density on Hotel.

(h) B: Density on Restaurant.

Figure 6.10: User Representation with Density of Different Methods on Yelp-Hotel and Yelp-Restaurant. The sub-figures (a), (b), (c), (d) contain the user representation information with the ground-truth labels, and the sub-figures (e), (f), (g), (h) show the density in the representation space. S refers to the social relation embedding-based method, and B refers to the behavior embedding-based method.

ratings. This embedding provides more comprehensive information for fraudulent sentiment analysis. For the cold-start problem, they efficiently infer the most probable representation of a new user in a closed-form solution according to the embedded user reviewing behavior. Four large real-word social media data sets demonstrate that the performance of the proposed methods is substantially better than the performance of state-of-the-art competitors.

Chapter 7

Conclusions and Future Directions

7.1 Conclusions

This thesis systematically investigates reliable sentiment analysis in social media, including paragraph sentiment analysis (in Chapters 3 and 4) and short text sentiment analysis (in Chapter 5), and fraudulent sentiment analysis, including supervised and unsupervised fraudulent sentiment analysis (in Chapter 6). Based on these investigations, this thesis proposes a serial of methods to achieve reliable sentiment analysis. Specifically, it studies the polarity-shift characteristics and non-IID characteristics in general paragraphs to capture the sentiment more accurately. It further models multi-granularity noise and sparsity in short text, which is the most common data in social media, for robust short text sentiment analysis. Finally, it tackles the uncertain credibility problem in social media by studying fraudulent sentiment analysis in both supervised and unsupervised scenarios. This thesis evaluates the performance and properties of the proposed reliable sentiment analysis methods by extensive experiments on large real-world data sets and demonstrates that the proposed methods are superior and reliable in social media sentiment analysis.

7.1.1 Paragraph Sentiment Analysis

This thesis first studies the reliable sentiment analysis for a paragraph in social media. Specifically, it investigates the polarity-shift factors and the non-IID characteristics, which dramatically increase the sentiment analysis complexity yet are hard to be captured. To tackle the polarity-shift problem, it proposes a multi-scale and hierarchical representation method to learn a more robust representation of textual data. To model the non-IID characteristics, it proposes a framework as well as an instantiation method, multi-scale and hierarchical deep neural network with an attention mechanism, to model and capture non-IID characteristics (i.e., couplings and heterogeneity). These studies build a foundation and provide useful tools for reliable sentiment analysis in social media.

7.1.2 Short Text Sentiment Analysis

This thesis further studies the short text sentiment analysis catering to the characteristics of textual data in social media. Specifically, it investigates multi-grain noise-tolerant patterns for sentiment analysis to tackle the noise and sparsity problem caused by a short text, which widely exists in reviews posted on social media. To capture these patterns, it proposes a breaking-gathering strategy with a bi-level multi-scale masked CNN-RNN network implementation, which embeds the most significant multi-grain noise-tolerant patterns in a text as the text's representation for reliable short sentiment analysis. This study provides specific tools for sentiment analysis of the most common textual data in social media. It complements the paragraph sentiment analysis to form the essential part of general reliable sentiment analysis in social media.

7.1.3 Supervised Fraudulent Sentiment Analysis

This thesis then studies the supervised fraudulent sentiment analysis method to detect fraudulent sentiment in social media. Specifically, it models user reviewing behavior by the co-occurrence relations between a user, an item, a review, and a rating. It then embeds this user reviewing behavior into a user-item-review-rating representation space, in which the representations of a user, an item, a review, and a rating will be similar if the user would like to write the review and give the score to the item. Furthermore, it further considers collaborative sentiment manipulation reflected by the co-reviewing relations between users and items. It embeds these co-reviewing relations and integrates these embeddings with the user-item-review-rating representation. Accordingly, this study generates a representation that comprehensively considers both individual and group factors for fraudulent sentiment analysis. It extends the horizon of reliable sentiment analysis in social media from handling high complexity and low quality to identify uncertain credibility.

7.1.4 Unsupervised Fraudulent Sentiment Analysis

The unsupervised fraudulent sentiment analysis studies how to detect fraudulent sentiment without the guide of human annotation, which is a common scenario in social media. It explores a self-supervision method to generate pseudo-labels from social media data. Specifically, it first adopts a dense-graph-mining technique to discover the collaborative fraudsters who are highly like to post fraudulent sentiment. It then treats the reviews posted by the discovered collaborative fraudsters contains pseudo-fraudulent sentiment, and the reviews posted by the others are honest to guide the adjusting of the learned user-item-review-rating representation. It also proposes a dynamic re-weighting strategy to increase dense-graph-mining precision based on the adjusted user-item-review-rating representation. By this means, it can achieve a practical unsupervised fraudulent analysis. To further tackle the cold-start problem in fraudulent sentiment analysis, this study constrains the user-item-review-rating representation to be an inferable representation where the representation of a new user can be inferred from the representation of its corresponding item, review, and rating. Overall, this study extends the fraudulent sentiment analysis from supervised learning to unsupervised learning, which significantly increases the practicality of the reliable sentiment analysis in social media.

7.2 Future Directions

The research in this thesis has many open problems and future opportunities. They include but not limited to: (1) exploiting more powerful support techniques for reliable sentiment analysis, (2) studying multi-granular reliable sentiment analysis, and (3) exploring the interpretability of reliable sentiment analysis.

7.2.1 Exploiting Other Support Techniques for Reliable Sentiment Analysis

This thesis builds a general framework for reliable sentiment analysis in social media. It also provides several instantiations, including analyzing paragraph sentiment with complex sentiment-polarity-shift and the non-IID characteristics, analyzing short-text sentiment with sparsity and noise, and analyzing fraudulent sentiment in both supervised and unsupervised fashion with the cold-start problem. However, it studies only a glance of support techniques for reliable sentiment analysis. For example, it analyzes the sentiment for paragraphs and short text relying only on deep neural networks. Although deep neural networks are promising for complex text representation, this technique requires a vast number of training data, which may not be available in a new social media community. This technique also needs a large computing capacity and has a large energy consumption, which does not suit for mobile platforms. Accordingly, the other techniques, such as shallow yet robust learning methods, should be further studied to support reliable sentiment analysis to be employed in a wide range of applications. For another example, this thesis builds the individual and group behavior model for fraudulent sentiment analysis based on co-occurrence embedding, in which, however, the temporal information is overlooked. Other support techniques are required to capture the temporal information for modeling behavior more precisely.

7.2.2 Studying Multi-Granular Reliable Sentiment Analysis

This thesis provides reliable sentiment analysis of the entire text. Although this study is valuable, different businesses may concern sentiment with different granularities, such as sentiment at aspect-level. For example, if a customer wants to buy a computer and reads a review for this computer, usually, she/he would like to know the sentiment to the computer performance instead of the sentiment to the shopping experience. As a result, the reliable sentiment analysis provided by this thesis cannot satisfy this customer's demand. To satisfy the demand for sentiment analysis at different granularities, the multi-granular reliable sentiment analysis should be further studied. The multi-granular reliable sentiment in social media has the following challenges: (1) the sentiment analysis granularity should match customer's intention, (2) the sentiment polarity of an aspect should be distilled from the entire text with complex interactions, and (3) the multi-granular reliable sentiment analysis lacks sufficient supervised labels. How to tackle these challenges is still an open problem.

7.2.3 Exploring the Interpretability of Reliable Sentiment Analysis

This thesis demonstrates the effectiveness and values of reliable sentiment analysis in social media. The provided reliable sentiment analysis methods can identify social media sentiment more precisely. To further extend the reliable sentiment analysis and widely apply it in real business, the interpretability of reliable sentiment analysis should be further studied. The interpretability will also provide more insight into reliable sentiment analysis. In real business, people not only requires a precise sentiment analysis result but also needs to know how to generate the result. For example, a customer may want to confirm whether she/he can trust the reliable sentiment analysis result by checking the interpretable model. For another example, a seller may desire the interpretable relations between different factors and sentiment polarity to her/his products for improving products to achieve positive sentiment. Accordingly, exploring the interpretability of reliable sentiment analysis is a promising opportunity.

APPENDIX A

APPENDIX

A.1 List of Notations

$\mathbf{U}$	user-message matrix (pp. 19)
$\mathbf{F}$	user-user matrix (pp. 19)
$\mathbf{A}_{sc}$	sentiment consistency matrix (pp. 19)
$\mathbf{A}_{ec}$	emotional contagion matrix (pp. 19)
$\mathbf{R}^{tt}$	microblog-microblog network (pp. 19)
$\mathbf{R}^{ww}$	word-word network (pp. 19)
$\mathbf{R}^{tw}$	microblog-word bipartite graph (pp. 19)
P	paragraph (pp. 47)
$\mathbb{P}$	paragraph space (pp. 47)
$\mathbb{E}$	paragraph embedding space (pp. 48)
n_s	the number of sentence (pp. 47)
s_i	the i-th sentence (pp. 47)
n_{wi}	the number of words in the i-th sentence (pp. 47)
$w_{i,j}$	the j-th word in the i-th sentence (pp. 47)

$E(\cdot)$	paragraph representation model (pp. 47)
$\mathbb{R}$	the real space (pp. 47)
$\mathbf{p}$	a paragraph vector (pp. 47)
n_f	the number of paragraph vector dimensions (pp. 47)
$o_{i,j}$	the sentiment polarity of the j-th word in the i-th sentence (pp. 47)
O_i	the sentiment polarity of the i-th sentence
$\mathbb{O}$	the polarity space (pp. 47)
$C(\cdot)$	sentiment classifier (pp. 47)
$\mathfrak{O}$	the distribution of the polarity of a set of paragraph (pp. 47)
$Div(\cdot\|\|\cdot)$	distribution divergence measurement (pp. 47)
$E_w(\cdot)$	the word-representation function (pp. 48)
$E_s(\cdot)$	the sentence-representation function (pp. 48)
$E_p(\cdot)$	the paragraph-representation function (pp. 48)
$\mathbb{W}$	the word space (pp. 48)
n_{ew}	the number of dimensions of word-representation space (pp. 48)
n_{es}	the number of dimensions of sentence-representation space (pp. 48)
n_{ec}	the number of dimensions of character-representation space
$\mathfrak{P}$	the distribution of a paragraph in space $\mathbb{P}$ (pp. 48)
$\mathfrak{E}$	the distribution of a paragraph in space $\mathbb{E}$ (pp. 48)
W	the weight of a nonlinear layer in a neural network (pp. 54)
b	the bias of a nonlinear layer in a neural network (pp. 54)
h_i	the output of a nonlinear layer in a neural network for the i-th sample (pp. 54)
x_i	the representation vector of the i-th sample (pp. 54)
u	the global memory of a context in the attention mechanism (pp. 54)
$exp(\cdot)$	the exponential function (pp. 54)

$c(i)$	the context set of the i-th sample (pp. 54)
α_i	the attention factor of the i-th sample (pp. 54)
x_i^*	the attentive representation of the i-th sample (pp. 54)
$\mathbf{E}_w$	the word embedding matrix (pp. 71)
$\mathbf{E}_c$	the character embedding matrix (pp. 71)
$\mathbf{T}_w$	the word transformation matrix (pp. 71)
$\mathbf{T}_c$	the character transformation matrix (pp. 71)
n_W	the number of unique vocabularies in a corpus (pp. 71)
n_C	the number of unique characters in a corpus (pp. 71)
n_w	the maximum number of words in a text (pp. 71)
n_c	the maximum number of characters in a text (pp. 71)
$\mathbf{M}$	the mask matrix (pp. 71)
u	a user (pp. 84)
t	a review (pp. 84)
d	an item (pp. 84)
r	a rating (pp. 84)
$\mathbf{u}$	a user representation (pp. 85)
$\mathbf{t}$	a review representation (pp. 85)
$\mathbf{d}$	an item represnetation (pp. 85)
$\mathbf{r}$	a rating representation (pp. 85)
$\mathbf{b}$	a reviewing behavior representation (pp. 85)
$s(\cdot)$	success rate of a behavior (pp. 88)

Bibliography

Arora, S., Liang, Y. & Ma, T. (2017*a*), A simple but tough-to-beat baseline for sentence embeddings, *in* 'International Conference on Learning Representations'.

Arora, S., Liang, Y. & Ma, T. (2017*b*), A simple but tough-to-beat baseline for sentence embeddings, *in* 'International Conference of Learning Representations'.

Asghar, M. Z., Khan, A., Ahmad, S., Qasim, M. & Khan, I. A. (2017), 'Lexicon-enhanced sentiment analysis framework using rule-based classification scheme', *PloS one* **12**(2), e0171649.

Baccianella, S., Esuli, A. & Sebastiani, F. (2010), Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining., *in* 'International Conference on Language Resources and Evaluation', Vol. 10, pp. 2200–4.

Belghazi, M. I., Baratin, A., Rajeswar, S., Ozair, S., Bengio, Y., Courville, A. & Hjelm, R. D. (2018), 'Mine: mutual information neural estimation', *arXiv preprint arXiv:1801.04062* .

Benevenuto, F., Magno, G., Rodrigues, T. & Almeida, V. (2010), Detecting spammers on twitter, *in* 'Collaboration, Electronic Messaging, Anti-abuse and Spam Conference', Vol. 6, p. 12.

Bengio, Y., Courville, A. & Vincent, P. (2013), 'Representation learning: A review and new perspectives', *IEEE transactions on pattern analysis and machine intelligence* **35**(8), 1798–1828.

Bespalov, D., Bai, B., Qi, Y. & Shokoufandeh, A. (2011), Sentiment classification based on supervised latent n-gram analysis, *in* 'CIKM', pp. 375–382.

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

Bollegala, D., Mu, T. & Goulermas, J. Y. (2016), 'Cross-domain sentiment classification using sentiment sensitive embeddings', *IEEE Transactions on Knowledge and Data Engineering* **28**(2), 398–410.

Bravo-Marquez, F., Mendoza, M. & Poblete, B. (2014), 'Meta-level sentiment models for big social data analysis', *Knowledge-Based Systems* **69**, 86–99.

Cao, L. (2013), 'Non-iidness learning in behavioral and social data', *The Computer Journal* **57**(9), 1358–70.

Chen, M. (2017), Efficient vector representation for documents through corruption, *in* 'International Conference of Learning Representations'.

Cortes, C. & Vapnik, V. (1995), 'Support-vector networks', *Machine learning* **20**(3), 273–297.

De Boom, C., Van Canneyt, S., Demeester, T. & Dhoedt, B. (2016), 'Representation learning for very short texts using weighted word embedding aggregation', *Pattern Recognition Letters* **80**, 150–156.

Deng, H., Lyu, M. R. & King, I. (2009), A generalized co-hits algorithm and its application to bipartite graphs, *in* 'Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 239–248.

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2018), 'Bert: Pre-training of deep bidirectional transformers for language understanding', *arXiv preprint arXiv:1810.04805* .

Dey, K., Shrivastava, R. & Kaushik, S. (2016), A paraphrase and semantic similarity detection system for user generated short-text content on microblogs, *in* 'Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers', pp. 2880–2890.

Diao, Q., Qiu, M., Wu, C.-Y., Smola, A. J., Jiang, J. & Wang, C. (2014), Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars), *in* 'Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 193–202.

Fan, W. & Gordon, M. D. (2014), 'The power of social media analytics', *Communications of the ACM* **57**(6), 74–81.

Feng, S., Banerjee, R. & Choi, Y. (2012), Syntactic stylometry for deception detection, *in* 'Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2', Association for Computational Linguistics, pp. 171–5.

Fernández, A. M., Esuli, A. & Sebastiani, F. (2016), 'Distributional correspondence indexing for cross-lingual and cross-domain sentiment classification.', *Journal of Artificial Intelligent Research* **55**, 131–63.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M. & Lempitsky, V. (2016), 'Domain-adversarial training of neural networks', *Journal of Machine Learning Research* **17**(59), 1–35.

Gao, M., Chen, L., He, X. & Zhou, A. (2018), Bine: Bipartite network embedding, *in* 'SIGIR', ACM, pp. 715–724.

Günnemann, S., Günnemann, N. & Faloutsos, C. (2014), Detecting anomalies in dynamic rating data: A robust probabilistic model for rating evolution, *in* 'Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM, pp. 841–50.

Hai, Z., Cong, G., Chang, K., Cheng, P. & Miao, C. (2017), 'Analyzing sentiments in one go: A supervised joint topic modeling approach', *IEEE Transactions on Knowledge and Data Engineering* **29**(6), 1172–85.

Hooi, B., Shin, K., Song, H. A., Beutel, A., Shah, N. & Faloutsos, C. (2017), 'Graph-based fraud detection in the face of camouflage', *TKDD* **11**(4), 44.

Hooi, B., Song, H. A., Beutel, A., Shah, N., Shin, K. & Faloutsos, C. (2016), Fraudar: Bounding graph fraud in the face of camouflage, *in* 'Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM, pp. 895–904.

Hovy, D. (2016), The enemy in your own camp: How well can we detect statistically-generated fake reviews–an adversarial study, *in* 'ACL', Vol. 2, pp. 351–356.

Hu, X., Tang, L., Tang, J. & Liu, H. (2013), Exploiting social relations for sentiment analysis in microblogging, *in* 'Proceedings of the Sixth ACM International Conference on Web Search and Data Mining', ACM, pp. 537–46.

Huang, M., Qian, Q. & Zhu, X. (2017), 'Encoding syntactic knowledge in neural networks for sentiment classification', *ACM Transactions on Information Systems* **35**(3), 26.

Ikeda, D., Takamura, H. & Okumura, M. (2010), 'Learning to shift the polarity of words for sentiment classification', *Transactions of the Japanese Society for Artificial Intelligence* **25**, 50–7.

Jia, C., Carson, M. B., Wang, X. & Yu, J. (2018), 'Concept decompositions for short text clustering by identifying word communities', *Pattern Recognition* **76**, 691–703.

Jian, S., Pang, G., Cao, L., Lu, K. & Gao, H. (2018), 'Cure: Flexible categorical data representation by hierarchical coupling learning', *IEEE Transactions on Knowledge and Data Engineering* **31**(5), 853–866.

Jiang, L., Yu, M., Zhou, M., Liu, X. & Zhao, T. (2011), Target-dependent twitter sentiment classification, *in* 'Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics', Association for Computational Linguistics, pp. 151–60.

Jiang, M., Cui, P., Beutel, A., Faloutsos, C. & Yang, S. (2014), Catchsync: Catching synchronized behavior in large directed graphs, *in* 'Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM, pp. 941–50.

Jindal, N. & Liu, B. (2008), Opinion spam and analysis, *in* 'Proceedings of the 2008 International Conference on Web Search and Data Mining', ACM, pp. 219–30.

Jindal, N., Liu, B. & Lim, E.-P. (2010), Finding unusual review patterns using unexpected rules, *in* 'Proceedings of the 19th ACM International Conference on Information and knowledge Management', ACM, pp. 1549–52.

Kennedy, A. & Inkpen, D. (2006), 'Sentiment classification of movie reviews using contextual valence shifters', *Computational Intelligence* **22**(2), 110–25.

Kim, Y. (2014), 'Convolutional neural networks for sentence classification', *arXiv preprint arXiv:1408.5882* .

Kingma, D. & Ba, J. (2014), 'Adam: A method for stochastic optimization', *arXiv preprint arXiv:1412.6980* .

Ku, L.-W., Liang, Y.-T. & Chen, H.-H. (2006), Opinion extraction, summarization and tracking in news and blog corpora, *in* 'AAAI Conference on Artificial Intelligence', pp. 100–7.

Le, Q. & Mikolov, T. (2014), Distributed representations of sentences and documents, *in* 'Proceedings of the 31st International Conference on Machine Learning', pp. 1188–96.

Li, F., Huang, M., Yang, Y. & Zhu, X. (2011), Learning to identify review spam, *in* 'IJCAI', Vol. 22, p. 2488.

Li, H., Chen, Z., Liu, B., Wei, X. & Shao, J. (2014), Spotting fake reviews via collective positive-unlabeled learning, *in* 'ICDM', IEEE, pp. 899–904.

Li, H., Chen, Z., Mukherjee, A., Liu, B. & Shao, J. (2015), Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns., *in* 'ICWSM', pp. 634–637.

Li, Q., Wu, Q., Zhu, C., Zhang, J. & Zhao, W. (2019), Unsupervised user behavior representation for fraud review detection with cold-start problem, *in* 'Pacific-Asia Conference on Knowledge Discovery and Data Mining', Springer, pp. 222–236.

Li, S., Lee, S. Y. M., Chen, Y., Huang, C.-R. & Zhou, G. (2010), Sentiment classification and polarity shifting, *in* 'Proceedings of the 23rd International Conference on Computational Linguistics', Association for Computational Linguistics, pp. 635–43.

Li, X., Li, C., Chi, J. & Ouyang, J. (2018), 'Short text topic modeling by exploring original documents', *Knowledge and Information Systems* **56**(2), 443–462.

Li, X., Wang, Y., Zhang, A., Li, C., Chi, J. & Ouyang, J. (2018), 'Filtering out the noise in short text topic modeling', *Information Sciences* **456**, 83–96.

Liang, S., Yilmaz, E. & Kanoulas, E. (2016), Dynamic clustering of streaming short documents, *in* 'Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM, pp. 995–1004.

Lim, E.-P., Nguyen, V.-A., Jindal, N., Liu, B. & Lauw, H. W. (2010), Detecting product review spammers using rating behaviors, *in* 'Proceedings of the 19th ACM International Conference on Information and knowledge Management', ACM, pp. 939–48.

Lin, Z., Feng, M., Santos, C. N. d., Yu, M., Xiang, B., Zhou, B. & Bengio, Y. (2017), A structured self-attentive sentence embedding, *in* 'International Conference of Learning Representations'.

Liu, B. & Zhang, L. (2012), A survey of opinion mining and sentiment analysis, *in* 'Mining text data', Springer, pp. 415–63.

Liu, S., Cheng, X., Li, F. & Li, F. (2015), 'Tasc: Topic-adaptive sentiment classification on dynamic tweets', *IEEE Transactions on Knowledge and Data Engineering* **27**(6), 1696–709.

Liu, S., Hooi, B. & Faloutsos, C. (2017), Holoscope: Topology-and-spike aware fraud detection, *in* 'CIKM', ACM, pp. 1539–1548.

Lochter, J. V., Zanetti, R. F., Reller, D. & Almeida, T. A. (2016), 'Short text opinion detection using ensemble of classifiers and semantic indexing', *Expert Systems with Applications* **62**, 243–249.

Luca, M. & Zervas, G. (2016), 'Fake it till you make it: Reputation, competition, and yelp review fraud', *Management Science* **62**(12), 3412–3427.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y. & Potts, C. (2011), Learning word vectors for sentiment analysis, *in* 'Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics', Association for Computational Linguistics, pp. 142–50.

Maaten, L. v. d. & Hinton, G. (2008), 'Visualizing data using t-sne', *Journal of Machine Learning Research* **9**(Nov), 2579–2605.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S. & McClosky, D. (2014), The stanford corenlp natural language processing toolkit., *in* 'ACL (System Demonstrations)', pp. 55–60.

Margarit, H. & Subramaniam, R. (2016), A batch-normalized recurrent network for sentiment classification, *in* 'Advances in Neural Information Processing Systems', pp. 2–8.

Meng, F., Gao, Y., Huo, J., Qi, X. & Yi, S. (2019), Neolod: A novel generalized coupled local outlier detection model embedded non-iid similarity metric, *in* 'PAKDD', Springer, pp. 587–599.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. (2013), Distributed representations of words and phrases and their compositionality, *in* 'Advances in neural information processing systems', pp. 3111–9.

Moghaddam, S. & Ester, M. (2010), Opinion digger: An unsupervised opinion miner from unstructured product reviews, *in* 'Proceedings of the 19th ACM international conference on Information and knowledge management', ACM, pp. 1825–8.

Mukherjee, A., Kumar, A., Liu, B., Wang, J., Hsu, M., Castellanos, M. & Ghosh, R. (2013), Spotting opinion spammers using behavioral footprints, *in* 'ACM SIGKDD', ACM, pp. 632–640.

Mukherjee, A., Venkataraman, V., Liu, B. & Glance, N. S. (2013), What yelp fake review filter might be doing?, *in* 'ICWSM'.

Ott, M., Choi, Y., Cardie, C. & Hancock, J. T. (2011), Finding deceptive opinion spam by any stretch of the imagination, *in* 'Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1', Association for Computational Linguistics, pp. 309–19.

Pan, S. J., Ni, X., Sun, J.-T., Yang, Q. & Chen, Z. (2010), Cross-domain sentiment classification via spectral feature alignment, *in* 'Proceedings of the 19th international conference on World wide web', ACM, pp. 751–60.

Pang, B., Lee, L. & Vaithyanathan, S. (2002), Thumbs up?: Sentiment classification using machine learning techniques, *in* 'The Conference on Empirical Methods on Natural Language Processing', Association for Computational Linguistics, pp. 79–86.

Passalis, N. & Tefas, A. (2017), 'Neural bag-of-features learning', *Pattern Recognition* **64**, 277–294.

Passalis, N. & Tefas, A. (2018), 'Learning bag-of-embedded-words representations for textual information retrieval', *Pattern Recognition* **81**, 254–267.

Pennington, J., Socher, R. & Manning, C. (2014), Glove: Global vectors for word representation, *in* 'Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)', pp. 1532–1543.

Perozzi, B., Al-Rfou, R. & Skiena, S. (2014), Deepwalk: Online learning of social representations, *in* 'Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 701–710.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. & Zettlemoyer, L. (2018), Deep contextualized word representations, *in* 'NAACL HLT', pp. 2227–2237.

Qian, Q., Huang, M., Lei, J. & Zhu, X. (2017), Linguistically regularized lstms for sentiment classification, *in* 'ACL', pp. 1679–1689.

Ralaivola, L., Szafranski, M. & Stempfel, G. (2010), 'Chromatic pac-bayes bounds for non-iid data: Applications to ranking and stationary β-mixing processes', *The Journal of Machine Learning Research* **11**, 1927–1956.

Rayana, S. & Akoglu, L. (2015), Collective opinion spam detection: Bridging review networks and metadata, *in* 'Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM, pp. 985–94.

Rayana, S. & Akoglu, L. (2016), Collective opinion spam detection using active inference, *in* 'ICDM', SIAM, pp. 630–638.

Ren, Y., Zhang, Y., Zhang, M. & Ji, D. (2016), Context-sensitive twitter sentiment classification using neural network, *in* 'Association for the Advancement of Artificial Intelligence', pp. 215–21.

Saif, H., He, Y., Fernandez, M. & Alani, H. (2016), 'Contextual semantics for sentiment analysis of twitter', *Information Processing & Management* **52**(1), 5–19.

Shi, T., Kang, K., Choo, J. & Reddy, C. K. (2018), Short-text topic modeling via non-negative matrix factorization enriched with local word-context correlations, *in* 'Proceedings of the 2018 World Wide Web Conference on World Wide Web', pp. 1105–1114.

Shi, Y., Suk, H.-I., Gao, Y. & Shen, D. (2014), Joint coupled-feature representation and coupled boosting for ad diagnosis, *in* 'CVPR', pp. 2721–2728.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P. & Van Der Linde, A. (2002), 'Bayesian measures of model complexity and fit', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **64**(4), 583–639.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. & Rabinovich, A. (2015), Going deeper with convolutions, *in* 'CVPR', pp. 1–9.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K. & Stede, M. (2011), 'Lexicon-based methods for sentiment analysis', *Computational Linguistics* **37**(2), 267–307.

Tan, C., Lee, L., Tang, J., Jiang, L., Zhou, M. & Li, P. (2011), User-level sentiment analysis incorporating social networks, *in* 'Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM, pp. 1397–405.

Tang, D., Qin, B. & Liu, T. (2015), Document modeling with gated recurrent neural network for sentiment classification, *in* 'Conference on Empirical Methods in Natural Language Processing', pp. 1422–32.

Tang, D., Qin, B., Wei, F., Dong, L., Liu, T. & Zhou, M. (2015), 'A joint segmentation and classification framework for sentence level sentiment classification', *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **23**(11), 1750–61.

Tang, D., Wei, F., Qin, B., Yang, N., Liu, T. & Zhou, M. (2016), 'Sentiment embeddings with applications to sentiment analysis', *IEEE Transactions on Knowledge and Data Engineering* **28**(2), 496–509.

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T. & Qin, B. (2014), Learning sentiment-specific word embedding for twitter sentiment classification, *in* 'ACL', Vol. 1, pp. 1555–1565.

Tang, J., Nobata, C., Dong, A., Chang, Y. & Liu, H. (2015), Propagation-based sentiment analysis for microblogging data, *in* 'Proceedings of the 2015 SIAM International Conference on Data Mining', SIAM, pp. 577–85.

Teng, Z., Vo, D. T. & Zhang, Y. (2016), Context-sensitive lexicon features for neural sentiment analysis, *in* 'EMNLP', pp. 1629–1638.

Turney, P. D. (2002), Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews, *in* 'ACL', pp. 417–424.

Vincent, N. & Ogier, J.-M. (2019), 'Shall deep learning be the mandatory future of document analysis problems?', *Pattern Recognition* **86**, 281–289.

Vo, D.-T. & Zhang, Y. (2015), Target-dependent twitter sentiment classification with rich automatic features, *in* 'International Joint Conference on Artificial Intelligence', pp. 1347–53.

Wang, D., Jiang, M., Zeng, Q., Eberhart, Z. & Chawla, N. V. (2018), Multi-type itemset embedding for learning behavior success, *in* 'Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining', ACM, pp. 2397–2406.

Wang, J., Wang, Z., Zhang, D. & Yan, J. (2017), Combining knowledge with deep convolutional neural networks for short text classification, *in* 'Proceedings of IJCAI', Vol. 350.

Wang, X., Liu, K., He, S. & Zhao, J. (2016), Learning to represent review with tensor decomposition for spam detection., *in* 'Conference on Empirical Methods in Natural Language Processing', pp. 866–75.

Wang, X., Liu, K. & Zhao, J. (2017), Handling cold-start problem in review spam detection by jointly embedding texts and behaviors, *in* 'ACL', Vol. 1, pp. 366–376.

Wang, Y., Huang, M., Zhu, X. & Zhao, L. (2016), Attention-based lstm for aspect-level sentiment classification., *in* 'The Conference on Empirical Methods on Natural Language Processing', pp. 606–15.

Wiebe, J. M., Bruce, R. F. & O'Hara, T. P. (1999), Development and use of a gold-standard data set for subjectivity classifications, *in* 'Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics', Association for Computational Linguistics, pp. 246–53.

Wu, F. & Huang, Y. (2016), Sentiment domain adaptation with multiple sources, *in* 'Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics', pp. 301–10.

Wu, F., Yuan, Z. & Huang, Y. (2017), 'Collaboratively training sentiment classifiers for multiple domains', *IEEE Transactions on Knowledge and Data Engineering* .

Xia, R., Xu, F., Yu, J., Qi, Y. & Cambria, E. (2016), 'Polarity shift detection, elimination and ensemble: A three-stage model for document-level sentiment analysis', *Information Processing & Management* **52**(1), 36–45.

Xia, R., Xu, F., Zong, C., Li, Q., Qi, Y. & Li, T. (2015), 'Dual sentiment analysis: Considering two sides of one review', *IEEE Transactions on Knowledge and Data Engineering* **27**(8), 2120–33.

Xia, R. & Zong, C. (2011), A pos-based ensemble model for cross-domain sentiment classification., *in* 'International Joint Conference on Natural Language Processing', pp. 614–22.

Xie, S., Wang, G., Lin, S. & Yu, P. S. (2012), Review spam detection via temporal pattern discovery, *in* 'Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM, pp. 823–31.

Xu, C., Zhang, J., Chang, K. & Long, C. (2013), Uncovering collusive spammers in chinese review websites, *in* 'Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management', ACM, pp. 979–88.

Xu, H., Wang, Y., Cheng, L., Wang, Y. & Ma, X. (2018), Exploring a high-quality outlying feature value set for noise-resilient outlier detection in categorical data, *in* 'CIKM', ACM, pp. 17–26.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A. J. & Hovy, E. H. (2016), Hierarchical attention networks for document classification., *in* 'The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies', pp. 1480–9.

Ye, J. & Akoglu, L. (2015), Discovering opinion spammer groups by network footprints, *in* 'ECML', Springer, pp. 267–282.

You, Z., Qian, T. & Liu, B. (2018), An attribute enhanced domain adaptive model for cold-start spam review detection, *in* 'COLING', pp. 1884–1895.

Zhang, X., Zhao, J. & LeCun, Y. (2015), Character-level convolutional networks for text classification, *in* 'Advances in Neural Information Processing Systems', pp. 649–57.

Zhu, C., Cao, L., Liu, Q., Yin, J. & Kumar, V. (2018), 'Heterogeneous metric learning of categorical data with hierarchical couplings', *IEEE Transactions on Knowledge and Data Engineering* .

Zhu, X., Guo, H., Mohammad, S. & Kiritchenko, S. (2014), An empirical study on the effect of negation words on sentiment, *in* 'ACL', pp. 304–313.

Zhuang, Y., Wang, H., Xiao, J., Wu, F., Yang, Y., Lu, W. & Zhang, Z. (2017), 'Bag-of-discriminative-words (bodw) representation via topic modeling', *IEEE Transactions on Knowledge and Data Engineering* **29**(5), 977–90.

Zuo, Y., Wu, J., Zhang, H., Lin, H., Wang, F., Xu, K. & Xiong, H. (2016), Topic modeling of short texts: A pseudo-document view, *in* 'Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining', ACM, pp. 2105–2114.