EMOTION ANALYSIS FROM TEXT

MINGLEI LI

Ph.D

The Hong Kong Polytechnic University

2018

The Hong Kong Polytechnic University
Department of Computing

Emotion Analysis from Text

Minglei Li

A thesis submitted in partial fulfilment of the requirements

for the degree of Doctor of Philosophy

August 2017

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____ (Signed)

Minglei Li (Name of student)

Abstract

With the rapid development of the Internet, a large amount of text data is generated every day. People express their feelings and emotions through the Internet. The study of emotion analysis from texts is essential to understand the sentiment and emotions of people in public, especially in various social media. This is a major step towards enabling machines to have affective intelligence.

This thesis focuses on emotion analysis from text, which studies four areas in emotion analysis, including (1) high-quality emotion corpus construction, (2) more comprehensive multi-dimensional emotion lexicon construction, (3) phrase level emotion analysis, and (4) fine grained emotion prediction based on event roles in context. The contribution mainly consists of five parts.

The first part is on high-quality emotion corpus construction. Due to prohibiting cost, earlier works on emotion corpus annotation have very limited success. Many works use automatic methods based on natural labels such as hashtags, which can be very noisy. In this thesis, a three-step selection framework is proposed to improve the quality of corpus using natural labels by filtering noises in microblog data. The framework includes both automatic noise removal and semi-automatic noise removal. Evaluation of this framework shows that the corpus acquired automatically is of high-quality with Kappa value reaching 0.92. It can reduce manual annotation workload by 45.5% with a relative improvement in quality by 23.0% in macro F-score.

The second part is on word level emotion analysis, namely multi-dimensional emotion lexicon construction which is more comprehensive and theoretically more sound. The biggest problem with emotion lexicons using discrete labels is its limited computability and extensibility. We propose to construct emotion lexicons based on multi-dimensional emotion model, such as Valence-Arousal-Dominance (VAD), Evaluation-Potency-Activity (EPA) using continuous values for each dimension. Then, a regression based method is

proposed to infer affective meanings of words from word embedding. Evaluation on various emotion lexicons shows that the proposed method outperforms the state-of-the-art methods on all the lexicons under different evaluation metrics with large margins. Comparing to other state-of-the-art methods, the proposed method also has a computational advantage. The emotion lexicons obtained using our methods are available for public access.

The third part investigates phrase level emotion analysis. Based on vector representations of words, compositional models can be used to infer vector representations of larger text units. In this work, we first investigate the effectiveness of different word representations in compositional models for phrases on a phrase sentiment analysis task. Representation models include multi-dimensional emotion lexicons, sentiment lexicon and word embedding. Results show that word embedding clearly outperforms special purpose emotion lexicons even though they are cognitively backed by theories. Secondly, we investigate how phrase embedding can be learned and thus emotions of phrases can be inferred from their embedding representation directly. A hybrid method is proposed to learn phrase embedding from both the external context as well as component words with a compositionality constraint in such a way to reduce the data sparseness problem and at the same time reduce the semantic problem for non-compositional phrases. Evaluation on four datasets shows that the performance of this hybrid method is more robust and can improve the phrase embedding.

The fourth part investigates fine-grained emotion analysis. Most studies on emotion analysis focus on the sentiment or emotion expressed by a whole sentence or document. In this work, a novel task is proposed to predict the emotion states of event roles in a specific event context, where an event role can be the subject, act and object involved in the described event. This is backed by cognition theory of Affective Control Theory (ACT) that emotion states are context dependent. The main idea is to use automatically obtained word embedding as word representation and use the Long Short-Term Memory (LSTM) network as the prediction model. Compared to the linear model used in ACT which uses manually annotated EPA lexicon, the proposed method outperforms the linear model and word embedding also performs better than EPA lexicon.

Together, our works show that (1) high-quality emotion corpus can be obtained through natural labels with proper noise elimination process; (2) provision of a sound and automatic method to obtain multi-dimensional emotion lexicons; (3) under different compositional models, word embedding representation performs better than other dimensional emotion representations; (4) both external context and component words are useful for learning the embedding of phrases; and (5) emotion under specific context can be inferred more effectively based on LSTM with word embedding. Word embedding as a general semantic representation is a promising word representation even in domain specific applications including emotion analysis.

List of Publications

- **Minglei Li**, Qin Lu, Yunfei Long and Lin Gui. "Inferring Affective Meanings of Words from Word Embedding." *IEEE Transactions on Affective Computing*, 8, no. 4 (October 2017): 443–56. 2017. doi:10.1109/TAFFC.2017.2723012.

- **Minglei Li**, Qin Lu, and Yunfei Long. "Phrase Embedding Learning Based on External and Internal Contexts with Compositionality Constraint." *Knowledge-Based Systems*, 2017. (Submitted)

- **Minglei Li**, Qin Lu, and Yunfei Long. "Are Manually Prepared Affective Lexicons Really Useful for Sentiment Analysis." *In Proceedings of the Eighth International Joint Conference on Natural Language Processing* (Volume 2: Short Papers) 2: 146–50. Taipei, Taiwan, 2017.

- **Minglei Li**, Qin Lu, Yunfei Long, and Lin Gui. "Affective State Prediction of Contextualized Concepts". *1st IJCAI Workshop on Artificial Intelligence in Affective Computing*, Melbourne, Australia, 2017.

- **Minglei Li**, Qin Lu, Yunfei Long. "Representation Learning of Multiword Expressions with Compositionality Constraint". *In proceedings of the 10th International Conference on Knowledge Science, Engineering and Management (KSEM)*, Melbourne, Australia, 2017.

- **Minglei Li**, Qin Lu, Yunfei Long, and Lin Gui. "Hidden Recursive Neural Network for Sentence Classification." *In Proceeding of International Conference on Computational Linguistics and Intelligent Text Processing (CICLING)*, Budapest, Hungary, 2017.

- **Minglei Li**, Da Wang, Qin Lu, and Yunfei Long. "Event Based Emotion Classification for News Articles." *In Proceedings of the 30th Pacific Asia Conference on Language, Information and Computation (PACLIC)*, Seoul, South Korea, 2016.

- **Minglei Li**, Yunfei Long, Qin Lu. "A Regression Approach to Valence-Arousal Ratings of Words from Word Embedding." *In Proceedings of International Conference on Asia Language Processing (IALP)*, Tainan, Taiwan, 2016. (Best Paper Award).

- **Minglei Li**, Qin Lu, Lin Gui, Yunfei Long. "Towards Scalable Emotion Classification in Microblog Based on Noisy Training Data." *In Proceedings of Chinese Computational Linguistics and Natural Language (CCL)*, Yantai, China, 2016.

- **Minglei Li**, Yunfei Long, Qin Lu. "Emotion Corpus Construction Based on Selection from Noisy Natural Labels". *In Proceedings of International Conference on Language Resources and Evaluation (LREC)*, Portorož, Slovenia, 2016.

Acknowledgement

The most important thing I learned during my PhD study is about how to obtain knowledge at the frontier of mankind. Exploring the unknown is definitely an interesting journey even though it can be bitter at times.

First of all, I would give my deepest appreciations to my supervisor, Professor Qin Lu, who is not only a good teacher but also a good friend. Pro. Lu not only guides me on how to conduct research but also on be a better person through setting examples by her own actions. I clearly remember many times that Pro. Lu helped me to polish my papers and thesis until late night, even after I went to sleep. Her insistence on consistency, coherence, and logic thinking will influence the rest of my life. In addition, I will always have fond memories of the badminton games we played together regularly.

Secondly, I would like to thank my BoE members, Professor Maosong Sun, Professor Ting Liu and Dr Grace Ngai, for their valuable comments on my thesis.

Thirdly, I am very grateful to Professor Maggie Wenjie Li, who gave me many thoughtful suggestions.

Fourthly, I would like to thank all my colleagues and friends. Thank Jian Xu and Tianyi Luo, whose diligence will always inspire me. Thank Yunfei Long, Lin Gui, and Jiyun Zhou for their assistance with my experiments and the valuable discussions. Thank Dan Xiong for her elaborate knowledge on daily life, which provides me great convenience. Thank Chengyao Chen for the inspiring discussions that help me to polish my papers. Thank Yanran Li for her profound knowledge on different research topics, which inspires me a lot. Her enthusiasm for research and life will always encourage me. Thank Yumeng Guo for the enlightening discussions that has deepen my understanding of machine learning. Thank Jiaxing Shen for his super-coding ability that helped me a lot. Thank Pingping Liu and Qingqing Zhao. Discussions between different disciplines really broadened my

horizon. Thank my lovely friends, Xue Pang, Shiying Lin, Ying Xin, Zhijian He and Kaining Yan. Your friendship made my life in PolyU a pleased one. The imageries of our fightings for the badminton champions will forever be imprinted in my mind.

Fifthly, I must give special thanks to my parents, Xinshan Li and Yuling Wang, and my elder sister and brother, Jing Li and Guanglei Li, for their loves, tolerance and unquestioned supports. Thanks for giving me such a wonderful and happy family.

Last but not least, I would like to thank my fiancee, Yaqiong Shen, for her support and understanding during my PhD study. Life would not be the same without her on my side.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Text has been one of the most important media for people to exchange information, express ideas, explain scientific discoveries and create stories, etc. The growing popularity of social media has fundamentally changed the web from a simple information dissemination platform to a more interactive and social network based platform not only for information exchange and sharing but also for personalized expressions of individual feelings. It sometimes also serves as a platform for online emotional support. As Professor R. W. Picard, a pioneer in affective computing, puts it, "Emotion pulls the levers of our lives, whether it be by the song in our heart or the curiosity that drives our scientific inquiry" [120]. Emotions expressed through webs, especially in different social media, can affect its readers in such an unprecedented speed and scale that sometimes it can have dire consequences. The ability to have emotion and the ability to express our feelings through written forms is one of the most important characteristics of human beings. It is one of the keys to distinguish human beings with other animals. It is also one of the keys to differentiate human beings from machines and robots.

The term emotion has many different definitions. In general psychology, Klaus R. Scherer gives a formal definition of **emotions** as *episodes of coordinated changes in several components (including at least neurophysiological activation, motor expression, and subjective feeling but possibly also action tendencies and cognitive processes) in response*

to external or internal events of major significance to the organism [131]. Emotion can be represented by **discrete emotion models** such as *anger, disgust, fear, joy, sadness, surprise* [32] or by **dimensional emotion models** using continuous values in every dimension such as the two-dimensional Valence-Arousal (VA) model [127]. Without going into formal definitions in psychology nor cognitive science, the term **affect** is used to describe a collection of feelings and other traits of human beings including emotion, mood, interpersonal stance, attitude, and personality traits [120, 131]. From this definition, it is easy to see that emotion is only one element of affect. Generally speaking, **affective computing** refers to the study of computational methods to assign computers with human-like capabilities regarding observation, interpretation, and generation of affect features [155]. **Emotion analysis** (**EA**) refers to computational methods to enable machines to recognize or generate emotions in a human manner. Even though emotion and affect are not the same in psychology, studies in affective computing are mostly focused on emotions. Thus, in this thesis, the terms *emotion analysis* and *affective computing* are used interchangeably. So are the terms *emotion* and *affective meanings*. A multi-dimensional emotion model is defined in the so called **emotion space** or **affective space**. The term **sentiment** is also used to describe people's feelings. However, sentiment is normally measured using polarities of positive and negative. In the emotion space, sentiment can be represented in a one-dimensional valence model in which polarity is indicated by a value either on the positive or the negative axis and the absolute value indicates intensity. Thus, in the emotion space, sentiment is simply a special one-dimensional emotion. **Sentiment analysis** aims to classify the sentiment polarity of a given piece of text. Thus, sentiment analysis can be viewed as a special kind of emotion analysis with focus on the valence dimension only. In this thesis, sentiment analysis is treated as a special kind of emotion analysis. Without loss of generality, sentiment lexicons and emotion lexicons are generally referred to as affective lexicons.

Emotions have been studied extensively in different disciplines such as psychology,

neuroscience, sociology, philosophy, medicine as well as computer science. Emotion analysis in computer science can be sub-divided into two main tasks: 1) emotion recognition and 2) emotion generation. **Emotion recognition** (ER) aims at identifying the emotions expressed in some media, such as images, text, video, audio, etc. Identifying emotions and changes to its recipient triggered by any of these media is also part of the emotion recognition work. On the other hand, **emotion generation** (EG) aims at enabling a machine to express emotions like a human, such as generating emotional faces [48], emotional voices [134], emotional text expressions and dialogs [190].

The types of emotion recognition are normally dependent on the kind of input data. Facial expression based ER analyses the expressed emotions using facial features through computer vision techniques [188]. Audio based ER analyses the expressed emotions using audio features and speech recognition techniques [27]. Electroencephalography (EEG) based ER analyses the emotions of a tested subject using brain wave features [61]. Gesture based ER analyses the conveyed emotions through body gesture features using computer vision techniques [42]. Text based ER analyses the emotions expressed in a piece of text [75]. Text based ER can also examine the power of words to its readers by predicting the emotional changes of either the subjects in text or the readers of the text [102].

This thesis focuses on emotion recognition from text only. Emotion generation is out of the scope of this study. Therefore, emotion analysis (EA) in this thesis refers to emotion recognition (ER) related tasks only.

Emotion analysis has many potential applications, such as in analysis of consumer's response to product, service, advertisement to help for future decisions [16, 114], recommendation for entertainments such as movies, books, music or pictures that are suitable for users' current mood [21], analysis of social responses to public events, such as a disaster, a war, a political event, news, etc [8], prediction of suicide tendency through social network [31], generation of appropriate responses with emotional recognition in dialog systems [123, 190], emotion analysis for an assistant system [86].

Emotion analysis has been studied for a long time and different methods are proposed. Earlier systems are mostly rule based [173, 28] using hand crafted rules and emotion lexicons. However, rule based systems suffer from scalability issues and knowledge acquired for one genre of text cannot be used by another genre easily. Newer EA systems mostly use machine learning (ML) methods [148, 174, 26]. Machine learning based methods mainly require three components. The first one is an emotion corpus used to train the ML models. The second one is some emotion-link knowledge base, often in the form of an emotion lexicon, which is used as an important feature for ML models [99]. The third one is the ML model itself. Even though there are many machine learning methods developed for EA, there are still some challenges.

1.1 Problem Statements and Research Objectives

The major problems and objectives that motivate this study will be given based on the three components in machine learning based EA.

Emotion corpus construction

Emotion corpus plays an essential role for training machine learning models. An emotion corpus consists of text annotated with emotion information. Previous methods for building emotion corpora either by **manual annotation** or by **distant supervision**. Manual annotation is time-consuming and hard to scale up. Many recent studies use distant supervision methods that make use of naturally annotated data from social media to automatically obtain labeled data to a great quantity [165, 152, 103]. Naturally annotated text features such as hashtags (the term inserted between two characters "#" by the author, called "topic" in Sina Weibo), emoticons and emoji characters in microblogs are automatically extracted from data and served directly as labels after some simple rule-based selection. The main problem with this method is that naturally annotated labels are noisy. Without appropriate methods to filter out noisy data, resulting corpora are less useful.

Objective 1: This work investigates how to make use of naturally labeled data effectively in order to obtain an emotion corpus with large quantity and at the same time, eliminate noisy data to obtain a high-quality emotion corpus.

Word level emotion analysis.

Word level emotion analysis is equal to emotion lexicon construction, which aims to assign affective information to words. Emotion lexicons are important resources for emotion analysis. An emotion lexicon consists of words annotated with emotion information, often referred to as **affective meaning** of words. For example, the word *party* should be annotated with the label *happiness* if a discrete emotion model is used. The affective meaning of a word can be represented using different methods. Earlier works in EA use discrete emotion labels to represent affective meanings of words, such as polarities *positive, negative* or multi-labels *happiness, sadness, anger, etc.* [145, 106, 144]. Other methods represent affective meanings by some more comprehensive multi-dimensional emotion models which represent the emotion in multi-dimensional space with each dimension using a continuous value. Commonly used models include the Valence-Arousal model (**VA**) [127], the Valence-Arousal-Dominance model (**VAD**) [17] and the Evaluation-Potency-Activity model (**EPA**) [51], etc. Sentiment is one dimension of the multi-dimensional models.

Compared to discrete emotion model or one-dimensional sentiment, multi-dimensional emotion model is more comprehensive because it can capture more fine-grained information. However, multi-dimensional emotion lexicons as natural language processing (NLP) resources are very limited because most available ones are based on manual annotation [17, 167, 183], which is not scalable and limits the use of multi-dimensional models in real applications. Automatic methods are proposed to help obtain a larger quantity of annotated affective resources. For example, word embedding based graph propagation method is proposed as an automatic method to predict the valence-arousal ratings from seed words [184]. However, words that have similar word embeddings may be associated

with different affective meanings. For example, *"father"* and *"dad"* have similar word embeddings, yet they are associated with different affective meanings; *"father"* is more formal and detached whereas *"dad"* is more personal and dear affectively.

Objective 2: This work explores more effective methods to automatically build emotion lexicons based on more comprehensive multi-dimensional emotion models.

Phrase level emotion analysis.

Phrase level emotion analysis aims to assign affective information to phrases. Previous machine learning based methods for emotion analysis are mainly based on manually defined features to obtain the feature representation of a target text. Commonly used features include bag-of-word (BoW), n-grams, emotion category counts, emoticons, emoji, hashtag, punctuations, text length, Part-of-Speech tagging (POS tagging), etc [3, 152, 146, 99]. However, feature engineering is time-consuming and domain dependent. In addition, such kind of feature is insufficient for phrases because phrases are too short to obtain engineered features. An interesting research question is: Can we directly learn the representation of a phrase for EA? Inspired by recent development in deep learning for NLP, which represent a word using a dense vector called word embedding [69, 44], composition models can be used to infer the representations of larger text units from representations of component words [97, 189, 185]. Various composition models are proposed, such as vector addition, element-wise vector multiplication, vector concatenation [97], tensor production [189], recurrent neural network [93], recursive neural network [139].

Objective 3: This work explores the effectiveness of different word representations in composition models to infer the representation of phrases for EA.

The next question is, however, what happens if some phrases are non-compositional? For a phrase like *battle cry*, compositional models can fail to learn its representation from its component words. In non-compositional cases, they can be treated as non-divisible

units, and their representations can be learned using **distributional models** that make use of the external context of these units based on the distributional hypothesis [49]. However, the biggest problem with the use of distributional methods is that it can suffer from data sparseness problem. The longer the non-divisible unit is, the severe the data sparseness problem is.

Objective 4: This work explores more effective phrase representation learning methods such that emotions can be inferred from the learned phrase representations.

Event role level emotion analysis.

For emotion analysis tasks, previous studies on emotion analysis aim to classify emotions expressed in a whole piece of text, such as sentences, paragraphs, or documents, etc.[105, 20]. Sometimes, the emotion that an author wants to express may necessarily be linked to an emotion state of either the subject or the object in the text. For human machine interaction, a machine needs to know more fine-grained information about the agent or patient in a piece of text that describes an event, such as the emotion of *"mother"* expressed in the sentence *"The mother hit the boy"* and the sentence *"The mother touched the baby"*.

Objective 5: This work explores methods to do fine-grained EA for the prediction of emotions of event roles.

1.2 Thesis Outline

Based on the identified problems and objectives, the thesis is organized into mainly four parts: emotion corpus construction, dimensional emotion lexicon construction, phrase level emotion analysis and fine-grained emotion analysis of event roles.

Chapter 2 introduces background knowledge related to this thesis, mainly including emotion models to represent emotions, word representation models that represent semantic

meanings of a word, composition models to obtain the representation of larger text units from word representations.

Chapter 3 introduces the proposed method for building a high-quality emotion corpus using naturally annotated labels. This part mainly focuses on how to make use of the naturally labeled data automatically from social media and at the same time try to eliminate noise to obtain high-quality data. The basic idea is to use a multiple-step method to first select high-quality naturally labeled data automatically and then, use semi-automatic method to select data in the remaining set for high-quality output. Result of this work is published in the paper [77].

Chapter 4 introduces the proposed regression-based method to construct multi-dimensional emotion lexicons using word embedding. The method is based on two assumptions: (1) different features in word embedding contribute differently to a particular affective dimension, and (2) one feature in word embedding also contributes differently to different affective dimensions. The proposed method treats word embedding as word features and learns meaning specific weights to each feature when mapping embedding to different affective dimensions. Result of this work is published in two papers [76, 81].

Chapter 5 introduces two works for phrase level emotion analysis. The first work investigates effective word representation models to be applied to compositional models for deriving the representations of longer text units such as phrases. Investigated word representations include multi-dimensional emotion lexicons and word embedding and investigated composition models include addition, multiplication, concatenation, and Long Short-Term Memory network (LSTM). Result of this work is published in the paper [78]. Results show that word embedding gives better results than specialized emotion lexicons because word embedding not only encode affective meanings, but also encode other useful semantic meanings. Following this conclusion, a hybrid model is proposed to learn embedding representation of phrases with consideration of both external context and internal component words. The learned embedding can be used to infer emotions of phrases using

the method proposed in Chapter 4. Result of this work is published in the paper [79].

Chapter 6 introduces a novel emotion analysis task to predict the emotions of different event roles in an event description. Prediction is based on LSTM trained on different roles of events in their proper context. Result of this work is published in the paper [80].

Chapter 7 concludes the thesis by summarizing the main contributions, limitations and future work on emotion analysis.

Chapter 2

Background

This chapter firstly introduces different emotion models. The second part gives an overview of the historical development of EA. The third part gives an overview of different word representations. The fourth part introduces different composition models that are used to compute the representations of larger text units based on word representations.

2.1 Emotion Models

Emotion, as the most complicated and fascinating part of human, has attracted many studies from different disciplines including the emotion mechanisms, how to recognize emotions and how to express emotions. How to represent emotion has been studied for a long time and different emotion models are proposed.

On the theoretical front on emotion understanding, Ortony et al. [111] propose a model of emotions, based on the appraisal theory in cognitive science, referred to as the OCC model (the abbreviation of the authors Ortony, Clore and Collins). In OCC model, emotions are classified into 22 types in a hierarchy according to the valenced reactions to different stimuli including reactions to events, agents (actions of agents), and objects. The OCC model analyzes emotions from the perspective of causes and how they can trigger certain emotions as reactions and how one emotion can be related to another. No personality and personal beliefs/values are considered in this model. Richard Lazarus et al. [67]

propose a unified view of appraisal and coping process model. It characterizes emotion as the result of some underlying mechanisms including both appraisal, which evaluates an organism's circumstances, and coping, which guides the response to this assessment. Appraisal and coping not only guide emotional behavior, but also play an important role in informing cognition, often in ways not considered by traditional models of intelligence. Affects are considered as the result of the appraisal of the environment and situationally interacted with goals, beliefs and intentions. Jonathan Gratch and Stacy Marsella [45] point out that the appraisal theories posit that events do not have significance in themselves, but only by virtue of their interpretation in the context of an individual's beliefs, desires, intentions and abilities. They further introduce the so called appraisal variables (also referred to as the appraisal dimensions) to characterize the individual variations. They attempt to propose a unified model that can simulate human emotional responses and also inform the debate on the general adaptive values of emotional reasoning. They further argue that beyond modeling the significance of events to one's self, appraisal variables also seem to play an important role in mediating social relationships. People readily appraise how events impact other individuals and use these appraisals to guide social actions. This is quite relevant to the study of emotion spread over the Internet.

Mahranian [90], in his Pleasure-Arousal-Dominance (PAD) emotional state model, makes distinctions of emotion states from temperament which he considers an individual's emotional traits measured as the average values regarding the three dimensions of emotions. He argues that dimensions of emotions include evaluation (pleasure), activity (activity) and potency (dominance) which are used to measure stimuli. Reactions to stimuli, which are called emotions, yield the same three factors. An individual's beliefs, generally termed as the value system, also place a very important role in each individual's reactions to the environment. Miller et al. [95] make use of the Personality, Affect, Culture (PAC) framework to simulate social agents which allow the social behavior to vary according to their personalities and emotions which, in turn, vary according to their motivations

and underlying motive control parameters.

Based on the proposed various emotion theories, emotions can be represented either by discrete categories or a set of values in continuous scales of some multi-dimensional space. In discrete representations, different categories are proposed. **Table 2.1** lists several discrete emotion models. Out of the seven discrete emotion models, Ekman's model with six categories is the most commonly used. Ortony's model is the most detailed. Xu's model differs from Ekman's by one additional *"like"*. And Xu's model is more commonly used in Chinese EA.

Table 2.1: List of popular discrete emotion models

Author	**Num**	**Basic Emotions**
Ekman[32]	6	anger, disgust, fear, joy, sadness, surprise
Parrot[116]	6	anger, fear, joy, love, sadness, surprise
Frijda[39]	6	desire, happiness, interest, sorrow, surprise, wonder
Plutchik[39]	8	acceptance, anger, anticipation, disgust, fear, joy, sadness, surprise
Tomkins[157]	9	anger, contempt, disgust, distress, fear, interest, joy, shame, surprise
Ortony[111]	22	fear, joy, distress, happy-for, gloating, hope, pity, pride, relief, resentment, satisfaction, etc.
Xu[175]	7	anger, disgust, fear, joy, like, sadness, surprise

On the other hand, the multi-dimensional emotion models represent emotions in multi-dimensional space with each dimension a continuous value. For example, in the valence-arousal model (VA) [127] as shown in Figure 2.1, each word is mapped to the affective space as a point in a two-dimensional space where valence indicates polarity and arousal indicates excitement; in the evaluation-potency-activity model (EPA) [112] as shown in Figure 2.2, each word is mapped to a three-dimensional space where evaluation indicates polarity, activity indicates excitement and potency indicates power; in the hourglass model of emotion [22], emotions are represented by four independent dimensions: pleasantness, attention, sensitivity and aptitude; other dimensional emotion models include the

four dimensions of evaluation-pleasantness, potency-control, activation-arousal, and unpredictability [37], and the three dimensions of serotonin, dopamine and noradrenaline based on neuroscience [89].

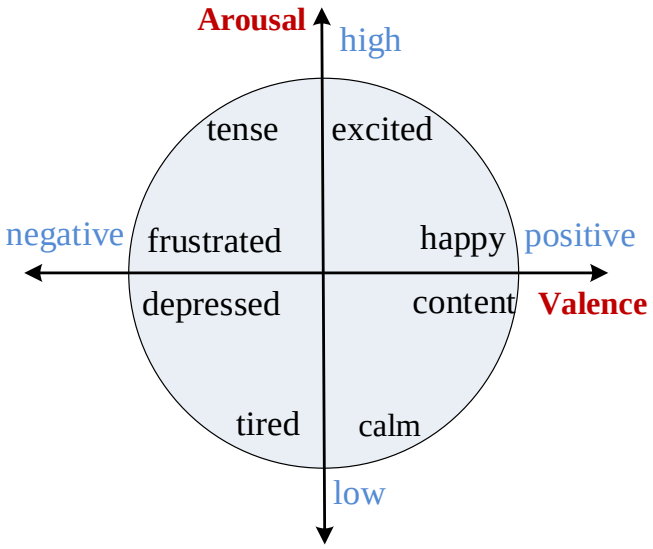


Figure 2.1: Two dimensional valence-arousal (VA) emotion model.

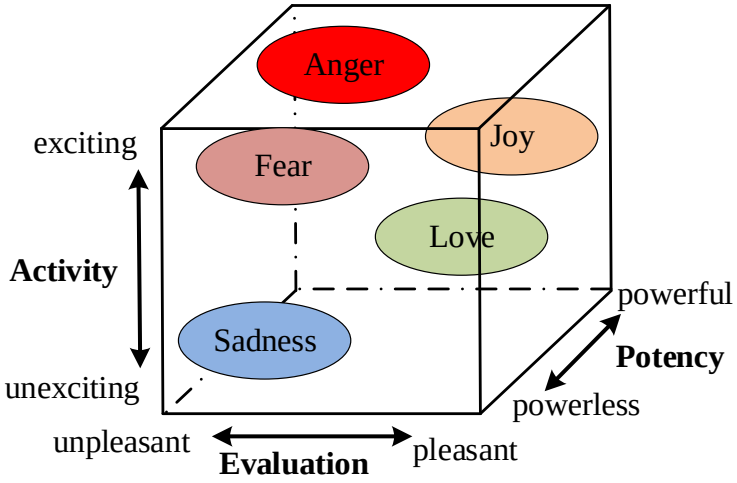


Figure 2.2: Three dimensional evaluation-potency-activity (EPA) emotion model.

Under the multi-dimensional emotion model, sentiment indicated by polarities can be viewed as a one-dimensional emotion model. For example, it is equal to the valence dimension in VAD or the evaluation dimension in EPA.

2.2 Overview of Emotion Analysis Tasks

One of the most important resources for emotion analysis is an emotion lexicon, which leads to research on emotion lexicon construction. Based on the emotion model used, it includes discrete emotion lexicon construction and multi-dimensional emotion lexicon construction. The methods for discrete emotion lexicon construction can be divided into manual annotations [175, 105, 106] and automatic methods [165, 104, 140]. The manually annotated discrete emotion lexicons include the Affect Lexicon [175], in which every word is labelled with one of 7 emotion categories together with strength of 5 levels, such as "谢世(die)" labelled with "悲伤(sadness)" with strength 5. This lexicon contains 27,446 Chinese words. Crowdsourcing is employed in [105, 106], which employ the Amazon's Mechanical Turk (AMT) to manually label the words with eight basic emotions (anger, anticipation, disgust, fear, joy, sadness, surprise, trust) and finally about 14,000 words and about 25,000 word senses with labelled emotions are obtained. It also deal with the polysemy problem through labelling several labels to that word, such as the word "accident" is labelled with "anger", "anticipation", "disgust", "fear", "joy", "sadness", "surprise", "trust". Automatic methods for building discrete emotion lexicon are mainly based on statistic information from large amount of texts from the Internet and social network. In [177], Yang et al. build emotion lexicon containing about 4,776 words from weblog using point-wise mutual information (PMI) of word and emotion pairs. In [144], Staiano et al. also use crowd source, which is free, to build emotion lexicon from news articles, compared to Amazon's Mechanical Turk which need to be paid. By employing the Rappler's Mood Meter, a small interface offering the readers the opportunity to click on the emotion that a given Rappler story makes them feel, 25.3K documents from rappler.com with emotion labelled are obtained. Different from previous annotation with only one discrete emotion label, the documents are labeled in all eight emotions with strength value, thus building a document-by-emotion matrix MDE. The document is then repre-

sented by word-by-document matrix MWD, and then a word-by-emotion matrix, called DepecheMood, is obtained through multiplication of MDE and MWD and some post-processing. This lexicon contains about 37k terms with each entry labelled with intensity of different categories, such as "kill#v" labelled with a predominant weight in AFRAID, AMUSED, ANGRY, ANNOYED, DON'T_CARE, HAPPY, INSPIRED, and SAD (0.23, 0.06, 0.21, 0.07, 0.05, 0.06, 0.05 and 0.27 respectively). The former lexicon are all from formal article such as news, blogs. Since microblog is becoming more and more important in people's social life, in which lots of informal expressions are used frequently, such as "u (you)", "thx (thanks)", Mohammad et al. construct an emotion lexicon from tweet [103]. The authors first obtain an emotion corpus-sentences labeled with an emotion-based on the hashtag when a tweeter proposes a tweet with a hashtag, then the unigram and bigram with emotion pairs are obtained through the PMI of the terms and related emotion labels. This emotion lexicon, called Hashtag Emotion Lexicon contains about 11,418 terms. For multi-dimensional emotion lexicons, the methods mainly includes manual annotation and automatic methods, which will be introduced in detail in Chapter 4.

Earlier works on emotion analysis are mostly conducted at the sentence level and document level. Research methods are developed in two stages. The first generation (before 2007) of EA is mainly rule-based, which is based on manually defined rules or linguistic patterns to analyze the emotion of a sentence. In [173], a set of emotion generating rules (EGR) are manually deduced, such as *"One may be HAPPY if he obtains something beneficial"*. Further more, the EGR can be divided into a domain-independent component such as *"obtain"* and a domain-dependent component such as *"something beneficial"*. Through hierarchical hyponym structure of a word, more EGR can be generated and used for emotion analysis. Chaumartin et al. [28] define rules based on common knowledge to analyze the emotions of news headlines. For example, the word inherited from *"Unhealthiness"* in WordNet will boost fear and sadness emotions. The authors also define the rule that the main subject in a sentence should weight more. Another kind of rule based

method is lexicon based, which adds the frequency or intensity of words in each emotion category and takes the word in a category with the maximum value as the final emotion label [144, 178]. Rule-based methods generally give higher precision. However, their fatal drawback is the coverage problem and the time-consuming rule definition.

The second generation (after 2007 to now) EA methods are mainly based on supervised machine learning (ML), whose framework is shown in **Figure 2.3**. This framework shows that ML pre-processes input text first and then converts it into a feature vector representation based on emotion lexicon and manually defined feature templates, such as the Bag-of-Word feature, the POS feature, the n-gram feature, the lexicon based feature [99, 163]. Based on the feature representation, a classifier or a regression model is trained on an emotion corpus and then used for emotion prediction.

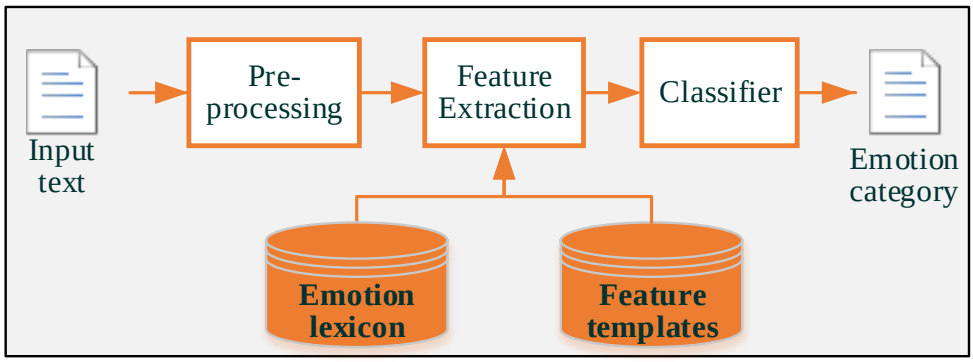


Figure 2.3: Machine learning framework for emotion analysis.

As an emotion corpus is a premise for supervised machine learning, emotion corpus construction becomes another research topic in emotion analysis community. The details of related work for emotion corpus construction will be introduced in Chapter 3.

Sentiment analysis, as a special type of EA, is also called opinion mining. As a field of study, sentiment analysis analyzes people's opinions, sentiments, appraisals, attitudes toward entities and their attributes expressed in written text [85]. The tasks in sentiment analysis can be categorized from different perspectives. Base on the type of targeted text

genre, the task can be divided into document level, sentence level, aspect level and word level. Document level analysis is to classify a whole document to either a positive or a negative sentiment [115]. Sentence level analysis is to determine whether a sentence expresses a positive or negative opinion. This is closely related to subjectivity classification which determines if a sentence express a subjective opinion (referred to as a subjective sentence) or a factual information (referred to as an objective sentence) [85]. Aspect level identifies which aspect is the target of the opinion and what is the opinion towards a specific target. Word level sentiment analysis is on sentiment lexicon construction, which assigns the polarity label or polarity with intensity to a given word. For example, *good, amazing* are positive and whereas *terrible, sad* are negative. According to [85], the main difference between emotion analysis and sentiment analysis is that the former detects multi-class emotions instead of identifying only the polarity of the target, which is a binary classification problem. Theoretically speaking, sentiment is a subset of emotions and all the tasks in sentiment analysis can be extended to emotion analysis. When using discrete emotion emotions, EA tasks become multi-class classification tasks from the corresponding binary classification tasks of sentiment analysis. When using multi-dimensional emotion models, sentiment is one dimension and the EA tasks become prediction the values in every dimension rather than only sentiment dimension.

Current works on emotion analysis are mostly at sentence level and document level using words as the basic semantic and affective units for aggregation and learning. However, phrases, as a semantically more meaningful units, may carry semantic information different from those of the component words and the affective meanings may also be different. For example, the phrase ”couch potato” is a non-divisible multi-word expression quite different from its component words of ”couch” and ”potato”. From study of literature, I have not found any systematic study on phrase level emotion analysis. There are only limited study in sentiment analysis, which analyze the polarity of phrases [172, 1].

In the following sections of this chapter, some models that will be used in the thesis

will be introduced.

2.3 Word Representations

How to find the most appropriate method to represent the meaning of a word has been the core issue both in the NLP community and the computational linguistic community. There are mainly five kinds of methods to represent a word: (1) symbolic representation, (2) manual feature based representation, (3) cluster based representation, (4) distributional representation, and (5) distributed representation. Each of them is introduced below.

1). Symbolic representation simply treats a word as a symbol and the word is transformed to a symbol ID, which is then transformed into a feature vector using a one-hot representation [158]. Feature vectors have the same length as the size of the vocabulary, and only one dimension is on. Bag-of-Word (BoW) feature representation of documents and sentences is typically represented symbolically based on one-hot vectors.

2). Manual feature based representation defines a word by a set of manually selected semantic features that indicate the different aspects of meanings as a set of semantic primitives. For example, in the common sense knowledge base ConceptNet [143], each concept is represented by manual defined features. For example, the word *apple* takes the value 1 under the features *can eat* and *is fruit* while takes the value 0 under the feature *can run*.

3). Cluster representation assigns each word a cluster class based on a hierarchical clustering algorithm, called Brown clustering, where each word is represented by the cluster it is assigned to [18]. In the clustering algorithm, the input to the algorithm is a text, which is a sequence of words $w_1, w_2, \cdots, w_n$. The output is a binary tree, in which the leaves of the tree are the words. Each internal node is interpreted as a cluster containing all the words in that subtree [83]. Each cluster can be encoded by a bit string, which can be used as the feature representations of the words.

4). Distributional representation is based on the distributional hypothesis that words

occur in similar context tend to have similar meanings [49]. Based on a given corpus, a word-context matrix M is constructed by setting the entry M_{ij} to 1 if word w_i and w_j co-occur in a context window. Then row i is the distributional representation of the word w_i. The dimension of the vector is equal to the vocabulary size, which is considered a high-dimensional representation [62]. Each entry is not limited to binary values. Co-occurrence frequency, mutual information, weighted mutual information, Term Frequency-Inverse Document Frequency (TF-IDF), point-wise mutual information (PMI) and the positive point-wise mutual information (PPMI) can all be used in the word-context vectors [62, 74]. As examples, Formula 2.1 gives the PMI definition of word-context co-occurrence matrix.

$$M_{i,j}^{PMI} = log\frac{P(w_i, c_j)}{P(w_i)P(c_j)} = log\frac{\#(w_i, c_j) \cdot |D|}{\#(w_i) \cdot \#(c_j)}. \tag{2.1}$$

Formula 2.2 gives the PPMI of word-context co-occurrence matrix.

$$M_{i,j}^{PPMI} = max(M_{i,j}^{PMI}, 0). \tag{2.2}$$

5). Distributed representation is also based on the distributional hypothesis. The main advantage is that it is a low-dimensional dense vector representation, also called **word vector** or **word embedding**, whose dimension is usually less than 1,000. Distributed representation can be obtained either through count-based methods or prediction-based methods [10]. A count-based method first builds the word-context co-occurrence matrix by the distributional representation. Matrix factorization is then performed on the co-occurrence matrix to obtain several low dimensional matrices and each row in one of the low dimensional matrix is used as word embedding. Different matrix factorization methods can be used including factorization into the multiplication of two matrix [119], Singular Value Decomposition (SVD) [135, 74], probabilistic matrix and tensor factorization [187], low rank approximation [82]. As an example, Formula 2.3 shows the SVD factorization on the

PPMI word-context co-occurrence matrix to obtain the word embedding,

$$M^{PPMI} = U\Sigma V^T. \tag{2.3}$$

Then, word vector for w_i is given by:

$$w_i^{SVD} = (U)_i. \tag{2.4}$$

The prediction-based methods do not need the word-context co-occurrence matrix. It directly learns word embedding through optimization on some neural network models based on the language model or word-context constraint in an unsupervised way. The language model based approach predicts the next symbol (usually words) given previous symbol sequence, which should make the sequence fluent. To be specific, given a sequence, s, of n words $s_1^n = w_1 w_2 \cdots w_n$ where the subscript 1 and superscript n of s indicates word 1 to word n, a language model decomposes the sequence s as the probability:

$$p(s_1^n) = \prod_{i=1}^{n} p(w_i | s_1^{i-1}), \tag{2.5}$$

where $p(w_i|s_1^{i-1})$ is the probability that word w_i occurs after the sequence s_1^{i-1}. Different neural language models are proposed to compute the conditional probability $p(w_i|s_1^{i-1})$. For example, in [12, 13], the approximation is based on the n-gram language model:

$$p(w_i|s_1^{i-1}) \approx p(w_i|s_{i-n}^{i-1}), \tag{2.6}$$

which takes a window of size n to approximate the whole sequence before word w_i. Then the probability $p(w_i|s_{i-n}^{i-1})$ is computed through a three layer neural network with word embedding as input. After training the language model by maximizing the likelihood of $p(s)$ over a large corpus, the word embeddings as the model parameters are also learned. Under this framework, different neural language models are proposed to compute the probability $p(w_i|s_{i-n}^{i-1})$. For example, in [98], Mnih et.al employ Restricted Boltzmann Machine

to model the probability $p(w_i|s_{i-n}^{i-1})$. In [30], Collobert et.al use the convolutional neural network to predict if the word in the middle of the input window is related to its context or not. In [58], Huang et.al combines local and global contexts to predict the score of next word based on previous word context window, where the global context is the weighted average of the word embedding of the whole document.

Previous word embedding learning models are computationally expensive. Mikolov et al. propose two simplified versions called the CBOW model (Continuous Bag-of-Words Model) and the Skip-gram model which are commonly used [94, 92]. The two models are shown in **Figure 2.4**.[1] The CBOW model predicts the target word given the context words in a window size. To be specific, the target word is predicted based on the sum of the vectors of the context words. On the other hand, the Skip-gram model predicts the context words in a context window size based on the target word. The main difference between the two models and the previous models is that CBOW and Skip-gram do not use a language model. They directly model the probability of the co-occurrence of a target word and its context word in a corpus. Thus, the computation cost of these two simplified models is much less than previous models.

Inspired by the idea of modeling relationship between a word and its context, other contexts are further explored, such as context words under specific syntactic dependency [72], context of words from different languages [36], context from knowledge base [166], neighbor context in semantic lexicon [35], substitute context [91], contrast context [110], path-based context [136], and morphological context [147, 35]. Further more, ensemble based methods are also proposed to make use of multi-view contexts [122, 60, 142]. For example, in [142], Speer et al. propose to combine word embedding from Skip-gram, word embedding from matrix factorization and word embedding from knowledge base ConceptNet to obtain ensemble word embedding. In [122], Rastogi et.al combine multi-view resources such as monolingual text from Wikipedia, word aligned bi-text, depen-

[1] Figure is modified from [92].

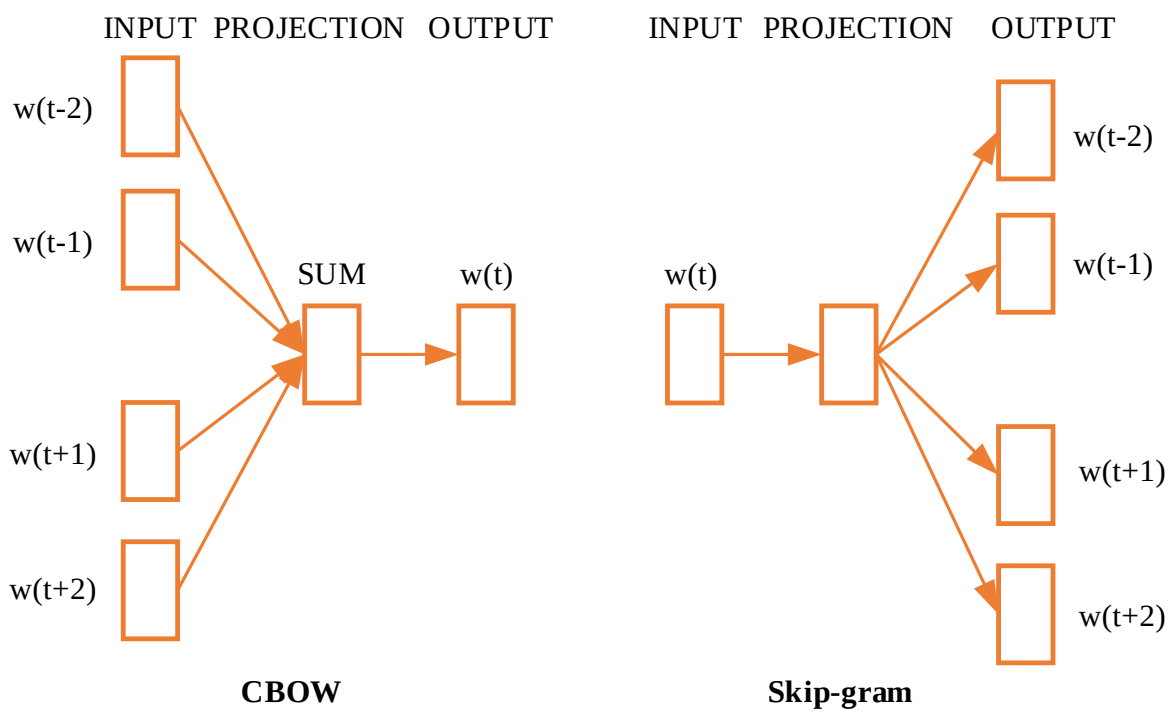


Figure 2.4: The framework of CBOW and Skip-gram for learning word embedding.

dency relations, morphology and Frame relations through Generalized Canonical Correlation Analysis (GCCA). In [29], word definition as an intrinsic view and context as an extrinsic view are used. Given current word, [29] maximizes the conditional probability of the context word and definition word, which is similar to [166] that maximizes the conditional probability of a target word given a current word, where the current word is from a knowledge base.

There is a close relationship between count-based methods and prediction-based methods. As proven in [73], the prediction-based method based on Skip-gram is implicitly factorizing a word-context co-occurrence matrix, whose cells are the PMI of the respective word and context pairs, shifted only by a global constant. However, since count-based methods need matrix factorization on a very large matrix (the same size as the vocabulary size), it costs more than prediction-based methods.

The difference between distributional representations and distributed representations is

the dimension size. The dimension of distributed representations is much smaller than distributional representations, which makes distributed representations widely used in deep learning based methods. Also, the use of dense vectors makes it possible for compositional methods to infer the representation of larger text units.

2.4 Compositional Models

One of the key element for natural language processing is to obtain a proper feature representation of the target text, such as words, phrases, sentences and documents. How to obtain the representation of larger text units is a key research problem. Here we focus on phrase and sentence representations. Methods for their representations can be mainly divided into three types: 1.) Feature template based methods; 2). Distributional methods; 3). Composition based methods.

Feature template based methods mainly construct the feature representation of target text by manually defined feature templates, such as word POS tagging, bag-of-word feature, n-gram feature, punctuation, negation, lexicon, ect. [104]. However, the manual feature engineering process is time-consuming and not scalable.

Distributional methods are inspired by the distributional methods for learning word representation as introduced in section 4.2.1. Distributional methods treat a sentence or a phrase as one single unit and directly learn their vector representation by using their context. For example, words occurrences or n-grams in documents are used as the context to learn sentence vector representations [68]. The surrounding sentences are directly used as context to learn sentence vector representations [63]. The surrounding words in a window are treated as context to predict the phrase vector representation [181]. However, distributional methods can suffer from data sparseness problem. The longer the text units are, the more severe the data sparseness problem is. The root of the problem is that a distributional method treats a sentence or a phrase as a non-divisible unit without consideration of

component words, which can contain meaningful semantic information.

Composition based methods, also referred to as **compositional methods**, are based on the **principle of compositionality** which states that the meaning of an expression is determined by the meanings of its component expressions and the rules used to combine them [38, 117]. The basic idea is to obtain the representation of a larger text unit from the representation of its component words through a computational approach. Let us use a two-word phrase as an example. Given a phrase p consisting of two words (w_1, w_2). The word embeddings of w_1, w_2 are $\vec{w}_m^1$, $\vec{w}_m^2$ respectively. The compositional methods can then obtain the representation of the phrase $\vec{p}$ based on the following function:

$$\vec{p} = f(\vec{w}_m^1, \vec{w}_m^2). \tag{2.7}$$

Partee [117] further suggests that the above principle should also take the role of syntax into consideration: The meaning of a whole is a function of its constituents as well as the syntactic rules to combine them. This can be modeled as:

$$\vec{p} = f(\vec{w}_m^1, \vec{w}_m^2, R), \tag{2.8}$$

where R is the syntactic relation between w_1 and w_2.

Lakoff [66] suggests that the meaning of the whole is greater than the meaning of its parts. The background information about the language and the knowledge related to the words should also be considered. So, the composition function should include an additional variable K, representing any knowledge associated with the composition process.

$$\vec{p} = f(\vec{w}_m^1, \vec{w}_m^2, R, K) \tag{2.9}$$

Compositional methods can obtain the representation of larger units recursively from the representations of its component words without any manual work. Consequently, compositional methods are widely used in NLP community now. In the following subsections, some commonly used composition models will be introduced.

2.4.1 Basic Composition Models

Several basic composition models are explored by [96, 97] based on the framework given in Formula 2.9 through some simplifications. Here only three basic linear models are shown. The **addition composition** model is defined as a linear composition:

$$\vec{p} = A\vec{w}^1 + B\vec{w}^2 = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} \vec{w}^1 \\ \vec{w}^2 \end{bmatrix}, \tag{2.10}$$

where A and B are the matrix to define the contribution of the component words. When A and B degrade to real numbers, the composition model degrades to a weighted addition model. When A and B further degrades to 1, the composition model degrades to a direct addition model in which both words contribute equally.

$$\vec{p} = \vec{w}^1 + \vec{w}^2. \tag{2.11}$$

The **multiplication composition** framework is defined by:

$$\vec{p} = C(\vec{w}^1 \bigotimes \vec{w}^2), \tag{2.12}$$

where $\bigotimes$ is the tensor product or outer product of vector $\vec{w}^1_m$ and $\vec{w}^2_m$, C is a tensor of rank 3 that maps the tensor product to the vector space of $\vec{p}$. By setting C to 1, a simplified version of multiplication composition model is produced as:

$$\vec{p} = \vec{w}^1 \circ \vec{w}^2, \tag{2.13}$$

where $\circ$ is the element-wise multiplication.

Another widely used composition model is the **concatenation composition** model which concatenates the vectors of the component words:

$$\vec{p} = [\vec{w}^1, \vec{w}^2]. \tag{2.14}$$

In addition to those basic composition models, more complex composition models are proposed by introducing different nonlinear transformations, which are widely used in different deep learning models [44]. Two non-linear neural network models, the recursive neural network (RecNN) and the recurrent neural network (RNN) will be introduced. Their relationship with the basic composition models will also be explained.

2.4.2 Recursive Neural Networks

Recursive neural network (RecNN) was first proposed by Socher for sentiment analysis [138, 137, 139]. Using the sentence *"The movie is cool"* as an example, the framework of RecNN is shown in Figure 2.5. When a word sequence is given to the RecNN model, it is first parsed into a binary parsing tree with each leaf node as a word. Then the representations of the parent nodes are obtained in a bottom-up fashion through a composition function f. After obtaining the representation of the root node, prediction can be performed on the root representation.

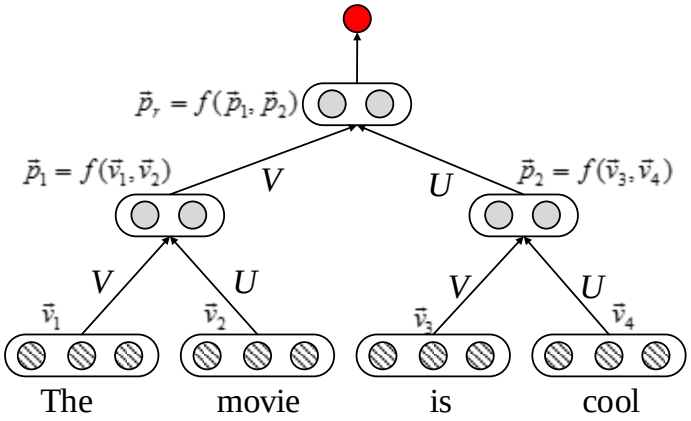


Figure 2.5: The composition model of RecNN for sentence representation learning.

Different composition functions f are proposed. Let $\vec{v}_p \in \mathbb{R}^d$ denote the vector of a parent node, and $\vec{v}_l$, $\vec{v}_r \in \mathbb{R}^d$ denote the vectors of the left and the right child nodes where d is the dimension size of the vectors. The original RecNN uses the following composition function [138]:

$$\vec{v}_p = \sigma\left(W\begin{bmatrix}\vec{v}_l \\ \vec{v}_r\end{bmatrix} + \vec{b}\right) = \sigma\left(\begin{bmatrix}W_l & W_r\end{bmatrix}\begin{bmatrix}\vec{v}_l \\ \vec{v}_r\end{bmatrix} + \vec{b}\right), \tag{2.15}$$

where σ is the element-wise activation function, which can be the sigmoid function, the tanh function or the ReLU function. This composition function actually concatenates the

two child vectors and then perform a matrix transformation with a bias, followed by a non-linear transformation. This can also be seen as the addition composition function in Formula 2.10 with a bias, followed by a non-linear transformation. The vectors of all words and matrices are treated as model parameters and learned during model training.

The second version of RecNN modifies f to:

$$\vec{v}_p = \sigma\left(W\begin{bmatrix} M_l\vec{v}_l \\ M_r\vec{v}_r \end{bmatrix} + \vec{b}\right), \tag{2.16}$$

where M_l and M_r are matrix that perform transformation on child vectors [137]. Formula 2.16 defines the so called Matrix Vector Recursive Neural Network (MV-RNN). Compared to the first composition function in Formula 2.15, the second version adds a matrix transformation before concatenating the two child vectors, which makes the parameters size much larger than the first model.

The third version of RecNN uses tensor product to capture more semantic interaction between two child words [139], referred to as the Recursive Neural Tensor Network (RNTN) defined as:

$$\vec{v}_p = \sigma\left(\begin{bmatrix} \vec{v}_l \\ \vec{v}_r \end{bmatrix}^{\mathrm{T}} V^{[1:d]}\begin{bmatrix} \vec{v}_l \\ \vec{v}_r \end{bmatrix} + W\begin{bmatrix} \vec{v}_l \\ \vec{v}_r \end{bmatrix} + \vec{b}\right), \tag{2.17}$$

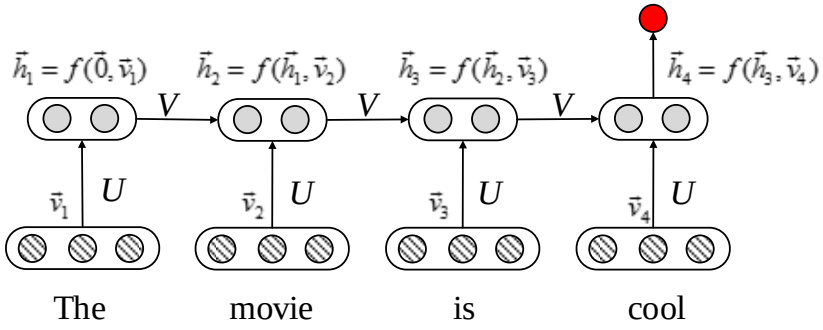
where $V^{[1:d]} \in \mathbb{R}^{2d\times 2d\times d}$ is the tensor that defines multiple bilinear to capture the dimension interaction between the two child words.

2.4.3 Recurrent Neural Networks

The framework of Recurrent Neural Network (RNN) is shown in Figure 2.6. Note that RNN can be viewed as a specific case of RecNN that the binary parsing tree in RecNN degrades to a linear chain.

In RNN, the second layer is called the hidden layer and each node is a hidden node.

Figure 2.6: The composition model of RNN for inferring sentence representation.



The composition function f of a hidden node i is modeled as:

$$\vec{h}_i = \sigma\left(V\vec{h}_{i-1} + U\vec{v}_i + \vec{b}\right) = \sigma\left(\begin{bmatrix} V & U \end{bmatrix}\begin{bmatrix} \vec{h}_{i-1} \\ \vec{v}_i \end{bmatrix} + \vec{b}\right), \tag{2.18}$$

where $\vec{h}_i$ is the representation of the hidden node i, which is composed from $\vec{h}_{i-1}$ and the current input node v_i. Figure 2.7 plots RNN in the form of a binary tree. It is now easier to see that RNN is a special case of RecNN. Each hidden node h_i in RNN can be treated as the internal parent node p_{i-1}. The only difference is that a "NULL" node with zero vector $\vec{0}$ is added on the far left. The advantage of RNN is that it is a simplified linear chain without requiring a paring tree.

However, a drawback with RNN is that it suffers from the gradient vanishing problem when the sequence is long. To overcome this problem, more complex composition model is proposed. The most widely used model is Long Short-Term Memory (LSTM) network [55]. The framework of LSTM is shown in Figure 2.8. In LSTM, the composition function f is a group of functions rather than a simple function. Each box as a node is called an LSTM cell and the input for cell t is the hidden representation $\vec{h}_{t-1}$ from previous cell $t-1$ and current word representation $\vec{v}_t$. An LSTM cell at position t consists of four parts: an input gate vector $\vec{i}_t$, a forget gate vector $\vec{f}_t$, an output gate vector $\vec{o}_t$, and a cell state vector $\vec{c}_t$. The output of each LSTM cell is defined by an output vector $\vec{h}_t$. These vectors are

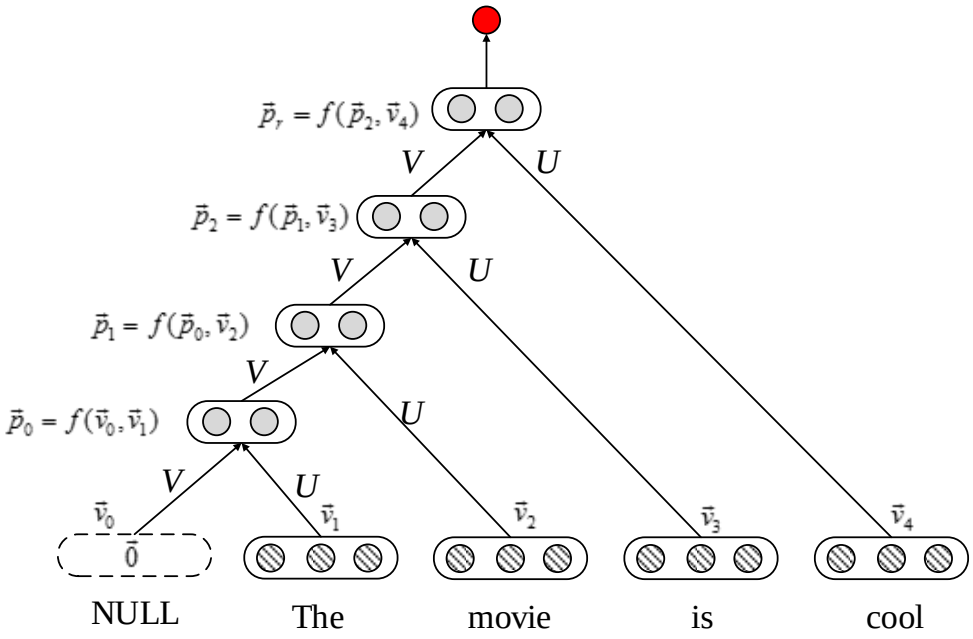
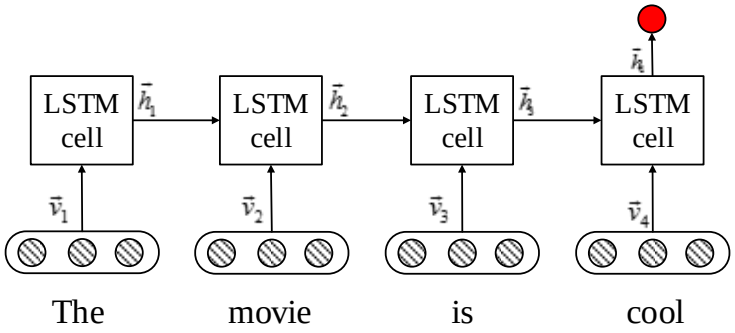


Figure 2.7: The composition model of RNN in the form of RecNN.

Figure 2.8: The composition model of LSTM.



defined as:

$$\vec{i}_t = \sigma(U_i\vec{x}_t + W_i\vec{h}_{t-1} + \vec{b}_i), \tag{2.19}$$

$$\vec{f}_t = \sigma(U_f\vec{x}_t + W_f\vec{h}_{t-1} + \vec{b}_f), \tag{2.20}$$

$$\vec{o}_t = \sigma(U_o\vec{x}_t + W_o\vec{h}_{t-1} + \vec{b}_o), \tag{2.21}$$

$$\vec{c}_t = \vec{f}_t \circ \vec{c}_{t-1} + \vec{i}_t \circ tanh(U_c\vec{x}_t + W_c\vec{h}_{t-1} + \vec{b}_c), \tag{2.22}$$

$$\vec{h}_t = \vec{o}_t \circ tanh(\vec{c}_t), \tag{2.23}$$

where σ is the sigmoid activation function, $\circ$ denotes Hadamard product, and $\vec{b}_i, \vec{b}_f, \vec{b}_o, \vec{b}_c$ are the bias. U_i, U_f, U_o, U_c, W_i, W_f, W_o, W_c are the model matrix parameters that are learned during training the model. Based on the current input $\vec{x}_t$ and the output $\vec{h}_{t-1}$ of previous cell, it computes the input gate vector $\vec{i}_t$, forget gate vector $\vec{f}_t$, output state vector $\vec{o}_t$. Then, the current output $\vec{h}_t$ is computed using the formula given below:

$$\vec{h}_t = \vec{o}_t \circ tanh(\vec{c}_t). \tag{2.24}$$

LSTM is good at remembering values for either long or short durations of time. If directly treating the input word vector as model parameters, this model can also be used to learn word embeddings. As a deep learning model, LSTM has been widely used in various NLP tasks, such as machine translation [149], language modeling [46], and sentiment analysis [151], etc.

2.5 Chapter Summary

In this chapter, the related background knowledge is introduced, including methods for emotion analysis, sentiment analysis, emotion models, word representations and composition models. These models and algorithms will be used in this thesis for different tasks. In the following chapters, the proposed methods will be introduced to overcome the related problems introduced in Chapter 1.

Chapter 3

Emotion Corpus Construction

One of the most important parts for supervised machine learning is annotated data. The availability of annotated data is vital to any supervised machine learning method to perform properly. Obtaining labeled data manually can be very time-consuming and noise prone especially for multi-class annotations or for subject related emotion annotation. This chapter reports a semi-automatic method to build emotion corpora.

Social media (such as Tweet, Sina Weibo) is a very important platform for people to share information. Huge amounts of data are generated in social media. Such data are becoming an important data source for machine learning researchers. One characteristic of social media data is that it contains **naturally annotated information**, namely **natural labels**, such as hashtags, emoticons and emoji characters. Thus, many studies take advantage of the social media by **distant supervision methods** to construct emotion copora automatically [9, 100, 165]. The basic idea is to extract naturally annotated data and select the appropriate annotations as emotion labels. However, these automatically obtained labels can be quite noisy. Take the following sentence as an example:

> “在你闲的时候，玩玩转发微博，未必不是一种乐趣！！！#无聊# (When you are not busy, playing with microblog retweet may be fun! #boring#)”,

which expresses “happiness” emotion if only text is examined. However, the additional negative hashtag “boring” is in conflict with the text. Thus, this hashtag should be con-

sidered as a noisy natural label. To show the adverse effect of noisy data, an experiment is conducted using a high-quality training data NLP&CC2013 [186] to train an emotion classifier, and naturally labeled data without noise filtering is inserted gradually to the training data to see the performance of the classifier affected by the naturally labeled data. NLP&CC2013 has a total of 4,000 sample blogs and 2,172 blogs contain emotion labels. The rest has neutral labels (namely has no emotions). **Figure 3.1** shows the experiment result. The x-axis is the number of added noisy data and the y-axis is the performance of the trained classifier. Note that the performance degrades continuously as more naturally annotated data are added. This indicates that if there is no appropriate data cleaning method, naturally annotated data can do more harm than good. Previous works on emotion corpus construction based on naturally annotated data use limited noise filter by some simple rules, such as removal of forwarding microblogs, microblogs containing URL, and microblogs less than four words, etc. [165, 101].

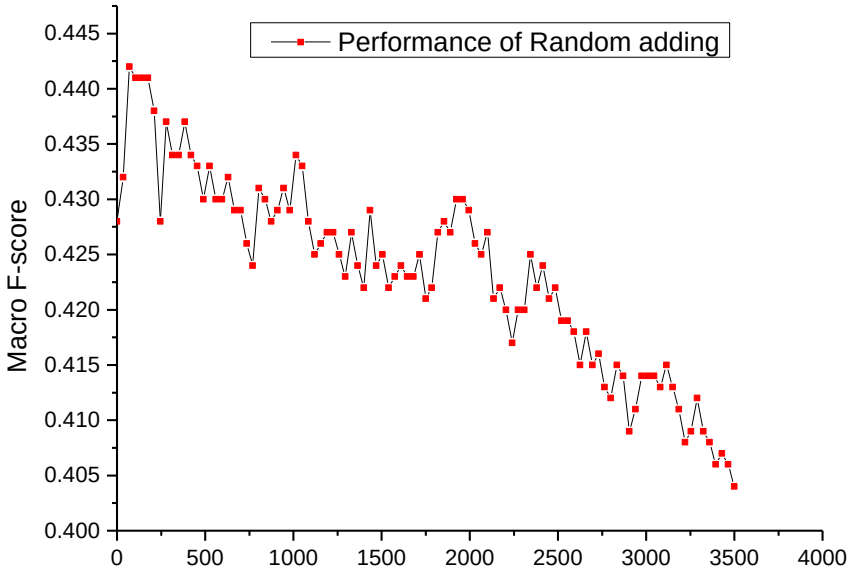


Figure 3.1: Performance of random adding

In this chapter, a more comprehensive noise removal method is proposed. The aim is to make use of the naturally labeled data effectively and at the same time try to elim-

inate noise to obtain high-quality data in large scale. The basic idea is to use a multiple stage method to first select high-quality naturally labeled data automatically and then, use experts to manually examine data in the remaining set to select more samples.

3.1 Related Work

Emotion corpus construction methods mainly include manual annotation by experts or crowdsourcing, and automatic methods. There are some emotion corpora based on manual annotation. For the English language, SemEval2007 [146] consists of only 1,250 news headlines labeled with the six Ekman emotion labels. The ISEAR dataset [130] consists of 7,666 sentences generated through questionnaires. In [5], Aman et al. construct an emotion corpus from web blogs and a total of 5,205 sentences are manually annotated based on the six emotion labels. Neviarouskaya et al. build an emotion corpus consisting of 700 sentences from a collection of diary-like blog posts [109]. This corpus is manually annotated with nine emotion labels with intensity. The Affect dataset [4] consists of more than 15,000 sentences from fairy tales with five emotion labels annotated manually. Because of the popularity of social media, emotion analysis of social media is attracting research attention. For example, 15,553 tweets annotated with 28 emotion categories through crowdsourcing via Amazon Mechanical Turk (AMT) are provided in [176].

For Chinese, the Ren-CECps (a Chinese emotion corpus developed by Ren-lab) [121] emotion corpus consists of 1,487 documents and 35,096 sentences from web blogs annotated with eight emotions. In addition, many emotion corpora are provided in shared tasks of emotion analysis. For example, a social media orientated Chinese corpus **NLP&CC2013**[1] [186] consists of 14,000 microblogs and 45,431 sentences from microblogs with 8 labels (including "none" label, meaning no emotion) through manual annotation. In NLP&CC2013, only 7,300 microblogs contain emotions and the size is still quite small. Further, NLP&CC2014[2]

[1] `http://tcci.ccf.org.cn/conference/2013/pages/page04_evares.html`

[2] `http://tcci.ccf.org.cn/conference/2014/pages/page04_eva.html`

is provided for the shared task of emotion analysis for social media in Chinese microblogs. This dataset contains 31,196 sentences annotated with the same labels as that of NLP&CC2013. In the shared task of emotional conversation generation of NLP&CC2017, a conversation corpus[3] annotated with six emotion labels (*Anger, Disgust, Happiness, Like, Sadness, Other*) has more than 1 million Weibo post-response pairs. There is also an emotion corpus for code-switch text that contains more than one type of languages. For example, 1,000 Weibo posts containing both Chinese and English are manually annotated with five emotion labels (*happiness, sadness, fear, anger and surprise*) [70]. All the above emotion corpora are labeled by discrete emotion categories. There is one Chinese emotion corpus that is based on the two dimensional valence-arousal model with about 2,009 Chinese sentences manually annotated and each dimension is annotated in the range of [1-9] [183]. The limitation of the manually annotated datasets is that they are hard to scale because of the cost of manual annotation.

Distant supervision method is used to automatically build emotion corpora based on naturally annotated labels. The news emotion categories labels are used as the emotion labels of news articles where the emotion category is given by readers after they read a news article [179]. Many news websites provide this kind of emotion tagging function, such as Yahoo News, Sina News[4]. Hashtags, emojis, emoticons, given by authors when they tweet, are used as the emotion labels to construct emotion corpora for social media text automatically [165]. Mohammad et al. build an emotion corpus from Tweet using hashtags, which contains about 21,000 tweets [103].

3.2 Selection Based Emotion Corpus Construction

In this study, a high-quality data selection framework is proposed to automatically or semi-automatically build emotion corpora by employing information from natural labels with

[3] `http://tcci.ccf.org.cn/conference/2017/taskdata.php`

[4] `http://news.sina.com.cn/society/moodrank/`

noise filtering. Commonly used natural labels include hashtags, emoticons, and emojis. The advantage of hashtag over emoticon and emoji is that it is straightforward to use them to search text in microblogs. Emoji and emoticon are not standardized and can not be searched easily. So hashtags are used as natural labels in this work. The whole construction procedure is shown in **Figure 3.2**. Firstly, a set of emotional seed words are selected manually, similar to other methods. Based on these seed words, microblogs are crawled to collect raw data. Then, a simple rule based preprocessing is performed on the raw data to get the preprocessed dataset **D0**. The preprocessed data **D0** then goes through a lexicon based selector to produce the first set of high-quality data, denoted as **H1**. The remaining unselected data **D1** goes through an SVM based selector to automatically produce the second set of high-quality data, denoted as **H2**. To further enlarge the corpus size, the remaining data **D2** can be manually selected to obtain the third set of high-quality data, denoted as **H3**. The remaining data D3 is discarded as noisy data. Previous distant supervision based methods on natural data selection mostly conduct Step 1, 2, 3 (marked by the blue box) to construct an emotion corpus. The proposed framework goes further to include Step 4, 5 and 6 (marked in the red box). Each step will be discussed in details in the following sections.

3.2.1 Hashtag Seed Selection and Data Crawling

To illustrate the proposed framework, Sina Weibo data is used as a demonstration. The framework itself is not proposed specifically for this data.

In Step 1, emotion labels are the same as the NLP&CC2013 corpus which takes seven emotion labels: *like, disgust, happiness, sadness, anger, surprise, fear*. A set of seed words are selected as hashtags for different emotion labels as listed in **Table 3.1**

Step 2 crawls the microblog data from Sina Weibo through Sina Weibo Topic API[5] based on the selected seed words. As a result, 173,958 microblogs with hashtags are

[5] `http://open.weibo.com/wiki/API`

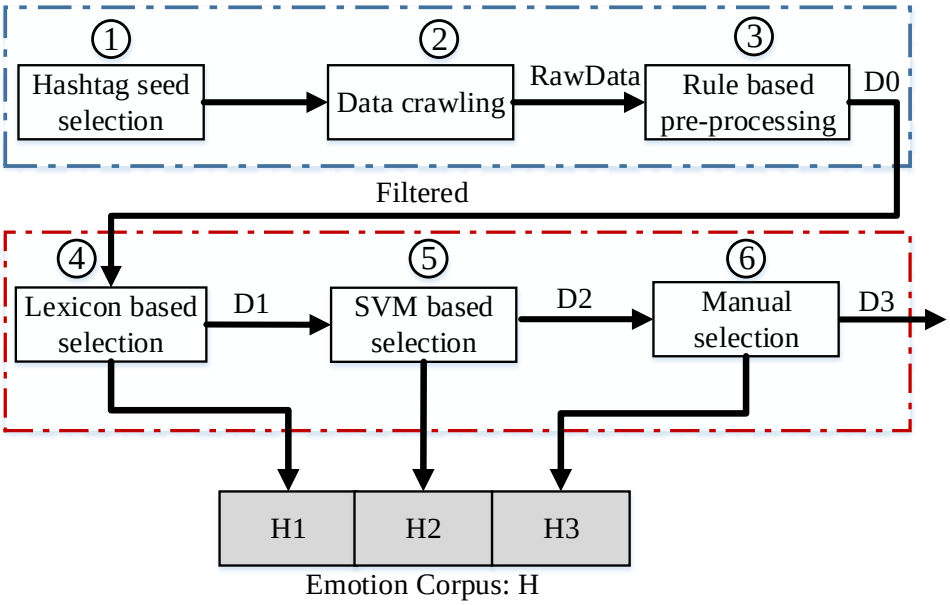


Figure 3.2: Emotion corpus construction framework

obtained. The raw data is denoted as **RawData**.

3.2.2 Rule Based Preprocessing

Since microblogs are from social media, the crawled data naturally contain noise. In Step 3, preprocessing is conducted on RawData based on the following removal rules.

1. Microblogs that contain less than 3 words excluding the hashtag and URL.
2. Duplicated microblogs in a discussion chain.
3. Microblogs that contain URL.
4. Forwarded microblogs.
5. Microblogs that are not in Chinese.
6. Microblogs that contain more than one hashtag.
7. Microblogs that contain quotes because such text is more likely to be dialogs, such as “讲个故事：“从前有个太监… … …” 有人耐不住问：“下面呢？” 继续

Table 3.1: Hashtag seed words

Emotion	**Seed word number and examples**
Like	9: “给力(helpful)” “可爱(lovely)” “奋斗(strive)” “喜欢(like)” “赞(appraise)” “爱你(love you)” “相信(believe)” “鼓掌(applaud)” “祝愿(hope)”
Disgust	9: “无聊(boring)” “烦躁(agitated)” “嫉妒(jealous)” “尴尬(embarrassment)” “讨厌(dislike)” “恶心(disgusting)” “怀疑(suspect)” “烦闷(bored)” “厌恶(disgust)”
Happiness	20: “快乐(happy)” “幸福(happy)” “哈哈(ha-ha)” “爽(so high)” “感动(moved)” “开心(joy)” “嘻嘻(happy)” “高兴(happy)” “亲亲(kiss)” “欢喜(happy)” etc.
Sadness	27: “伤不起(can’t bear the hurt)” “郁闷(sadness)” “哭(cry)” “失望(disappointed)” “心塞(heart hurt)” “难过(sadness)” “思念(long for)” etc.
Anger	27: “妈的(fuck)” “无语(speechless)” “气愤(angry)” “恼火(anger)” “tmd” “你妹的(your sister)” etc.
Surprise	16: “神奇(miracle)” “惊呆了(shocked)” “不可思议(inconceivable)” “天哪(my god)” “大吃一惊(shocked)”
Fear	11: “害怕(fearful)” “紧张(nervous)” “心慌(nervous)” “害羞(shy)” “(embarrassed)” etc.

讲故事：“下面？没了啊... ... ” (Translation: Let me tell a story: ”long time ago, there is a eunuch...”. Some impatient person interrupted: ”What’s next?”(In Chinese, ”What’s next” also means: ”what is down there”), I continued my story. ”What’s next?”, ”no more!”) This is a Chinese joke on eunuchs. There are many microblogs filled with ambiguity and sarcasm like this.

8. Microblogs that contain abnormal hashtags, such as “神奇(amazing)”, which is the name of a movie. This is performed through a manual review of raw data.

9. Microblogs whose hashtag is not at the start or the end of the text. This is because some hashtags are used as a part of the content and do reflect the emotion of the text. For example, in the sentence “我#讨厌#小孩子，很讨厌很讨厌(I #dislike# children, very very dislike children)”, “dislike” is the content of the text, if treat it

as a hashtag and remove it from the text, the text is incomplete.

Also, all traditional Chinese text are converted to simplified Chinese. Through the above preprocessing, 48,245 (27.77%) microblogs are kept as **D0** out of 173,958, indicating that the filtering percentage is 72.23%. The hashtags as natural labels in the cleaned microblogs, denoted as the D0 dataset, are converted to the corresponding emotion labels given in Table 1 and the text is segmented by the Chinese segmentation tool Jieba[6] for further processing.

3.2.3 Lexicon Based Selection

Since natural labels are noisy as stated in introduction part, Step 3 uses a lexicon based method for verification. This method selects natural labels based on an emotion lexicon counting strategy. The algorithm is given in Algorithm 3.1. Given a piece of segmented text, the numbers of words that actually occur in each emotion category in the emotion lexicon are counted and the emotion category with a maximum count is used as the predicted emotion label of this piece of text. If the counts for different categories are equal, all of them are used as predicted labels. Then, if the original natural label is in the predicted labels, the natural label is used and the sample is added to the selected high-quality dataset H1. Otherwise, they will be included in the remaining set for further processing. The emotion lexicon used in this step is DUTIR[7] and a collection of popular Internet words. The DUTIR lexicon contains about 27,000 words that are manually annotated based on seven discrete emotion labels the same as NLP&CC2013. After lexicon based selection, 14,197 microblogs with high-quality labels are obtained as H1. The remaining 34,108 microblogs, denoted as **D1**, are used by Step 5 for further processing.

Lexicon based selection helps to identify text that contains explicit emotional words or emotional affinity words. Text that expresses emotion through word combination cannot

[6] `https://github.com/fxsjy/jieba`

[7] `http://ir.dlut.edu.cn`

Algorithm 3.1 The Lexicon Based Algorithm

Input :

$W = [w_1, w_2, \cdots, w_m]$: Segmented text with m words.
y_o: the natural label of text W from the hashtag.
$Y = y_1, y_2, \cdots, y_n$: The emotion label set.
$S = [s_1, s_2, \cdots, s_n]$: Emotion lexicon. s_i is a word list with emotion y_i.

Output:

H: The selected high-quality dataset.
$D1$: The remaining dataset for further processing.

Procedure:

1. Set $C = [c_1, c_2, \cdots, c_n]$ where $ci = 0$
2. for w in W:
3. for s_i in S:
4. if w in s_i:
5. $c_i = c_i + 1$
6. $c_{max} = argmax(C)$
7. for c_i in C:
8. if $c_i = c_{max}$:
9. add y_i to y
10. if y_o in y:
11. add W to H
12. else:
13. add W to $D1$

be classified by a lexicon, such as "今天我这里又没有水了*(Today there is no water again in my place)*" which expresses disappointment (as a form of sadness) through the combination of "no" and "water". To cover these complex cases, machine learning based selection is used in Step 5.

3.2.4 SVM Based Selection

Step 5 uses a Support Vector Machine (SVM) [150] based selection method. The basic idea is that a classifier is trained first based on available high-quality emotion corpus. Then, this classifier is used to predict the remaining data from Step 4. If the predicted label is the same as the original natural label, it is regarded as a high-quality label and is put in H2. Otherwise, it is put in D2 for further processing. Features used in SVM include BoW with stop words removed. SVM is implemented with Liblinear [33]. Training data is from NLP&CC2013 introduced in Section 3.1, which contains 14,000 microblogs. We only use the 4000 microblogs of the training data part in NLP&CC2013 because the test data will be used for evaluation later. After SVM based selection, 7,228 more microblogs

are obtained as H2, and the remaining 26,820 are put in D2.

3.2.5 Manual Selection

The manual selection step is optional. If there are no annotators, this step can be skipped. As will be indicated later, this step can help to improve recall for the set of illustrative data used in this chapter. If using annotators is too costly, one straightforward solution is to craw more data. If annotators are available, manual annotation can be done using the remaining D2 data. To ensure quality, annotators are not allowed to see the natural labels. Each sample in the final annotated data should have only one label. If an annotator considers some samples to have multiple possible labels, they can put down two labels and only the one that matches the natural label (by an automatic process after manual labeling) will be selected. Otherwise, the sample will be discarded.

In this work, one trained annotator is asked to do manual annotation. The annotation rules are as follows:

1. Only consider emotion of the author through his or her written content.

2. If the author describes something with positive or negative words, the labels should be either "like" or "disgust".

3. Each microblog may contain several sentences, the emotion of the microblog should be for the whole section of the text. For example, the microblog *"今天出门上班摔了一跤，不过还好碰到了个大帅哥把我带到了公司(Today I fell down when I went to work. Fortunately, a handsome guy gave me a ride to my office.)"*, which expresses "sadness" in the first part and "happiness" in the second part. However, the whole text should be labeled as "happiness".

4. One microblog can be annotated with at most two emotion labels and the one that matches the natural label will be used.

5. For text that is meaningless without looking at context out of a single blog post, it is discarded.

6. For text that does not contain emotion, the label should be "none". This sample should be discarded.

Since only one annotator is trained to perform the annotation, there is a chance that the manual label is incorrect. There is also a chance that the natural label is incorrect. However, the chance of both labels are incorrect is much lower. So only those samples whose natural labels are consistent with the manual labels are extracted by a program automatically. Finally, 18,236 microblogs are obtained in H3. The remaining 8,584 microblogs form D3 which is considered as noisy data. In other words, a further 17.8% (8584/48,245) data are screened out.

Table 3.2: Example samples selected and remainders by different steps.

Step	**Selected**		**Remainder**		
	Text	Hashtag	Text	Hashtag	Prediction
Lexicon	明天期末考试了，真心希望能逆袭成功！！*(The final exam will come tomorrow and I do hope I can win the battle!!)*	like	别人心情不好至少还有酒能麻痹自己放空脑袋*(They at least have wine to paralyze themselves to empty the head when in bad mood.)*	sadness	disgust
SVM	作业好多啊！！！！！肿么做得完啊*(So many homework!!!!! How can I finish it)*	sadness	想回家了*(I want to go home)*	sadness	likeness
Manual	这就是高考以后的生存状态么。。。。*(Is this the living state after College Entrance Examination....)*	disgust	李丹，我真心爱你！永不变心！*(Li Dan, I truly love you! I will never change my heart)*	sadness	happiness

Table 3.2 lists some examples under different steps. The text listed under the **Selected** Column with text and their corresponding Hashtag which means that the original emotion

hashtag is the same as predicted emotion label. The **Remainder** column contains examples filtered out by their corresponding steps. **Hashtag** under **Remainder** is the original emotion hashtag and **Prediction** is the predicted emotion label, which is different from **Hashtag**. For example, the sentence *"想回家了(Want to go home)"* in the SVM row is not recognized by the lexicon based selection because no words in the sentence is in the emotion lexicon. SVM based selection gives the "like" label whereas the original hashtag is "sadness".

3.3 Analysis of Acquired Corpus

After the completion of all six steps, the final cleaned emotion corpus H has 39,661 samples comprised of three parts. H1 and H2 are obtained automatically. H3 is obtained manually. The distribution of the three parts is shown in **Table 3.3**. Note that automatically obtained data accounts for more than 54.1% in the final selected corpus. This is translated into a reduction of manual work by 44.5% out of the D0 set. If manual annotation is not feasible, this means that to obtain the same amount of annotated data, about 1.8 times of the automatically crawled data should be used.

Table 3.3: Proportioin distribution of obtained corpus

	H1	**H2**	**H3**	**H**
Size	14,197	7,228	18,236	39,661
Percentage (%)	35.74	18.35	45.91	100.00

The distribution of emotion classes in H is shown in Table 3.4. It shows that "sadness" and "happiness" have more samples whereas "surprise" and "fear" are much less. This is consistent with the manually annotated corpus NLP&CC2013. This distribution also demonstrates the intrinsic data imbalance problem in EA.

Table 3.4: Emotion distribution

Emotion	Number	Percentage (%)
sadness	14052	35.43
happiness	9959	25.11
disgust	4876	12.29
anger	4562	11.50
like	4540	11.45
surprise	1011	2.55
fear	661	1.67
sum	39661	100.00

3.3.1 Quality Analysis

To evaluate the quality of the emotion corpus, a sampling of about 5% for each H1, H2, and H3 in H is done with the same label proportion given in Table 3.4. Then, another annotator is asked to manually annotate the three sample sets based on the same rules described in section 3.2.5. anually annotated labels are compared to the natural labels (namely the gold labels) and the Kappa values are calculated. The result shown in Table 3.5 indicates that the Kappa value achieves 0.941, 0.926 and 0.812 for H1, H2 and H3, respectively. This indicates that the label quality in this work is much higher than the Kappa value of 0.713 in NLP&CC2013. The relatively high Kappa values indicate that the proposed method is quite effective in obtaining high-quality data.

For those text with inconsistent labels, analysis to the data reveals that sometimes the natural labels are even more reasonable than the manual ones. For example, in the sentence *"今天放假了, 我会想念你们的！(Holiday begins today, and I will miss you!)"*, the natural label and the lexicon based label are both *"sadness"* whereas the manual label is *"like"*. In fact, this sentence means that the author feels sad for not being able to see his/her friend, so he/she puts a "sadness" hashtag. On the other hand, there are also samples in our answer set where the natural labels are not reasonable. For example, in the sentence *"移动总是乱扣费！(China Mobile always charges ridiculous fees)"*, the natural label

and the SVM based label are both *"sadness"* whereas the manual label is *"anger"*. In this sentence, "anger" is more reasonable. These two examples actually show that not only computer aided methods may introduce noise, manual annotation can also give incorrect answers. More samples of the constructed corpus can be found in **Appendix A**.

Table 3.5: Kappa value of automatically selected label

Data	**Size**	**Sample Size**	**Kappa**
H1	14,197	700	0.941
H2	7,228	400	0.926
H3	18,236	900	0.812

To prove that the acquired data is a useful resource, an experiment is conducted to evaluate the use of baseline emotion corpora and different parts of the corpus produced by this work in an emotion classification task. The baseline corpora include (1) D0, which is obtained using simple rule based filtering; (2) the training set of NLP&CC2013. For the NLP&CC2013 dataset, about half of them are with label "none", which cannot be obtained through the hashtag, so the "none" label is discarded both in NLP&CC2013's training and testing data. The resulted training and testing sample sizes are 2,172 and 5,182 respectively. The other datasets examined in this experiment also include H1, H2, H1&H2, H as well as H&NLP&CC2013 (the combination of H and NLP&CC2013). These corpora are tested by training the same classifier using these dataset and the trained classifiers are tested on the NLP&CC2013 testing dataset. Since the classifiers are the same, the assumption is that if the quality of a corpus is higher, the performance of the classifier trained on it should be better.

The features are simple bag-of-word frequency. The classifier is Liblinear with the default parameter and metric is macro precision, recall and F-score, which can be calculated as follows:

$$Macro\_Precision = \frac{1}{n}\sum_{i}\frac{\#system\_correct(emotion = i)}{\#system\_proposed(emotion = i)} \tag{3.1}$$

$$Macro\_Recall = \frac{1}{n}\sum_{i}\frac{\#system\_correct(emotion = i)}{\#gold(emotion = i)} \tag{3.2}$$

$$Macro\_F - score = \frac{2 \times Macro\_Precision \times Macro\_Recall}{Macro\_Precision + Macro\_Recall} \tag{3.3}$$

where n is the number of emotion labels, #gold(emotion=i) is the number of samples whose gold emotion label is i, #system_correct(emotion=i) is the number of samples whose predicted label is the same as gold label i, #system_proposed(emotion=i) is the number of samples whose predicted label is i.

The results are shown in Table 3.6. Column "Size" is the corpus size, "MP" is macro precision, "MR" is macro recall, "MF" is macro F-score, and "Improve D0%" is the relative improvement over D0 in macro F-score. H&NLP&CC2013 is the union of the two sets H and NLP&CC2013. As expected, D0 obtained by simple filtering rules achieves the worst result which serves as the baseline. H1 alone has comparable performance to D0 although its size is less than one-third of D0. Also H1 performs worse than NLP&CC2013, its precision is quite high compared to NLP&CC2013. Its problem is the recall as it is lexicon-selection based samples. H2, with a very small size, is worse than D0 by more than 10%. However, when H1 and H2 are merged, it performs much better than NLP&CC2013 and has a relative improvement of 10.76% in macro F-score to NLP&CC2013 and 18.47% to D0. This implies that H1 and H2 are complementing to each other even though the samples identified by H2 is only about half of H1. When all three components are merged as H, the performance is even better. The precision, recall and F-score have an improvement over D0 by 23.9%, 22.3% and 23.0%, respectively. This indicates the effectiveness of the proposed noise removal framework. When H and NLP&CC2013 are used together to have the largest training set, precision does not improve over H whereas recall is only improved marginally. This indirectly demonstrates that the quality of data obtained using the proposed framework is even better than the manually obtained data of NLP&CC2013.

Table 3.6: Performance of different corpora on NLP&CC2013 test dataset

Training data	**Size**	**MP**	**MR**	**MF**	**Improve D0%**
Simple(D0)	47,243	0.3870	0.3024	0.3395	0.00
NLP&CC2013	2,172	0.3734	0.3534	0.3631	6.95
H1	14,197	0.4338	0.2930	0.3498	3.03
H2	7,228	0.2954	0.3142	0.3045	-10.31
H1&H2	21,425	**0.4841**	0.3440	0.4022	18.47
H	39,661	0.4793	0.3698	0.4175	22.97
H&NLP&CC2013	41,833	0.4762	**0.3732**	**0.4185**	**23.27**

3.3.2 Noisy Data Analysis

This section focuses on analyzing the 8,584 microblogs in D3, the noisy data after Step 6, to evaluate the noise level in the natural labels. Since D3 is obtained automatically by a consistency check program, the inconsistency can only result from two factors: (1) natural labels in D3 contain noise, and (2) manual annotation is incorrect. The first case with their labels is denoted as L1 and the second case with their labels is denoted as L2. In this experiment, another trained annotator is asked to annotate every sample in D3 based on the same annotation rules in Section 3.2.5, and the labels are denoted as L3. Then the voting strategy is used to determine the final label L. If L1 equals to L3, it is denoted as a high-quality label, a noisy label otherwise. The annotation result is shown in Table 3.7, where the entries represent the percentage of L1=L3, L2=L3 and others, respectively. The total noisy labels account for 72.5% in D3, which converts to 12.9% in dataset D0. This means that about 12.9% of microblogs after simple rule-based processing contain noise.

Table 3.7: Statistics of the noisy data

	L1=L3	**L2=L3**	**Others**
Percentage (%)	27.5	44.3	28.2

3.4 Chapter Summary

This chapter presents a framework for noise removal on a corpus obtained by distant supervision methods. The framework can automatically or semi-automatically construct a high-quality emotion corpus from a noisy corpus using natural hashtag labels.

An experiment on microblogs indicates that out of 173K raw microblogs, about 48K are filtered as candidates for high-quality data. Using the proposed framework including the optional manual annotation, 39k microblogs are selected to form the final high-quality corpus with Kappa value reaching 0.92 for the automatically selected part and over 0.81 for the manually selected part. The proportion of the automatically selected part is 54.1% and the manual part is 45.9%, which translates to a reduction of about 44.5% workload compared to the manual workload for acquiring high-quality data. Experiment on a classifier using this corpus as training data shows that it achieves better results compared to the classifiers trained on the manually annotated NLP&CC2013 training corpus and on the corpus obtained by simple rule based filtering. As a result of this work, 39,661 Chinese microblogs with high-quality emotion labels are obtained and are made available[8], which can be used for Chinese emotion analysis and is about five times of NLP&CC2013 if only samples with emotion labels are considered.

One drawback of this method is that the obtained corpus has no neutral label, namely text without emotion labels. However, this may be settled by adding crawled sentences that do not have hashtags contained in the seed hashtags subjected to some additional filtering works. This will be one possible future research direction.

[8] `https://yunfeilongpoly.github.io/Team_resource.html`

Chapter 4

Emotion Lexicon Construction

Another important resource for emotion analysis is a comprehensive emotion lexicon, in which words are annotated with affective meanings. Assigning affective meanings to words can be considered as word level emotion analysis. As introduced in Section 2.1, the affective meaning of a word can be represented using different emotion models. Earlier works represent affective meanings of words by discrete emotion labels, such as *positive, negative, happiness, sadness, anger* [145, 106, 144], etc. Another method is to represent affective meanings by the more comprehensive multi-dimensional representation models , such as the valence-arousal-dominance model (VAD) [127] and the evaluation-potency-activity model (EPA) [51]. Theoretically speaking, discrete emotion labels can always be mapped to certain points in a multi-dimensional affective space [20]. Sentiment indicated by polarities can be viewed as a one-dimensional emotion model. For example, it is equal to the valence dimension in VAD or the evaluation dimension in EPA.

Compared to discrete emotion labels or one-dimensional sentiment, multi-dimensional affective representation is more comprehensive because it can capture more fine-grained information compared to the discrete and the one-dimensional models. According to the Affective Control Theory (ACT), each concept in an event has a transient affective meaning which is context dependent in addition to cultural, behavior and other background information [52]. Multi-dimensional models allow for more interaction between a sequence

of words so that more context information can be included in emotion analysis of text. For example, the same noun *champion* may have different affective state in two different events: *The little boy defeated the champion* and *The champion defeated the little boy*. The difference of the affective states cannot be inferred through single sentiment dimension but it can be distinguished through multi-dimensional EPA emotion lexicons based on the ACT [52].

However, multi-dimensional emotion lexicons as NLP resources are limited because most available ones are based on manual annotation, such as the ANEW lexicon of VAD [17], the extended ANEW lexicon [167], the Chinese valence-arousal lexicon [183], and the EPA lexicon [51]. Obviously, manual annotation is not scalable and it limits the use of multi-dimensional models in real applications. Only if automatic methods can be used to learn the affective representations of words, the more comprehensive multi-dimensional models can have a wider practical use.

Word embedding based graph propagation method is used as an automatic method to predict the valence-arousal ratings from seed words [184]. However, word embedding is normally trained to obtain the general meaning of words, which can include denotative meaning, connotative meaning, social meaning, affective meaning, reflected meaning, collocative meaning and thematic meaning [71]. In other words, directly computing word similarity captures the general meanings of words rather than the affective meanings specifically. Words that have similar denotative meanings may be associated with different affective meanings. For example, *"father"* and *"dad"* have the same denotative meaning, yet they are associated with different affective meanings: *"father"* is more formal and detached whereas *"dad"* is more personal and dear affectively.

In this chapter, a regression method is proposed to infer various affective meanings from word embedding based on the assumption that different features in word embedding contribute differently to a particular affective dimension and one feature in word embedding also contributes differently to different affective dimensions. This method treats word

embedding as word features and learns meaning specific weights to each feature when mapping embedding to different affective dimensions. Consequently, the method learns one regression model for each affective dimension based on the seed words to predict the affective meaning of a new word provided that its word embedding be available.

4.1 Related Work on Emotion Lexicon Construction

Based on the selective emotion models, emotion lexicons are built either using a discrete emotion model or a dimensional emotion model. This chapter will only focus on dimension based lexicons. Since sentiments can be described by a one-dimensional model, methods for obtaining sentiment lexicons are also included. Theoretically speaking, methods to obtain a sentiment lexicon can be extended to obtain other affective dimensions in multi-dimensional models.

Emotion lexicons can be obtained either by manual annotation or automatic methods. Manual annotation can obtain high-quality lexicons. Manually annotated sentiment lexicons include the General Inquirer (GI) [145], MPQA[172], the twitter sentiment lexicon based on crowdsourcing [104, 125], VADER based on crowdsourcing[59], etc. Manually annotated emotion lexicons based on discrete emotion models include the DULTIR emotion lexicon in Chinese [175] which has been used in Chapter 3 for the lexicon based selection algorithm, the English emotion lexicon obtained through crowdsourcing [106] which contains about 17,000 words. Manually annotated multi-dimensional emotion lexicons include ANEW, CVAW, DAL, EPA and ANGST, among others. The ANEW lexicon based on the VAD model [17] contains 1,034 English words. The extended ANEW lexicon contains about 13,965 English words annotated through crowdsourcing. The CVAW lexicon based on the VA model contains 1,653 traditional Chinese words annotated in the valence and arousal dimensions [183]. The Dictionary of Affect in Language (DAL) lexicon annotated in the dimensions of pleasantness-activation-imagery contains 8,742 terms

[169]. The EPA lexicon annotated in the evaluation-potency-activity dimensions [52] contains about 4,505 English terms. The ANGST lexicon annotated in the valence-arousal-dominance-imageability-potency dimensions contains 1,003 German words [132].

Automatic methods to obtain emotion lexicons are focused mainly on the sentiment dimension because current research works are mostly on sentiment analysis [101, 84, 23, 47]. In terms of methodology, there are mainly three approaches. The first approach uses statistical information between a target word and the polarity annotated seed words. For example, sentiment polarity intensities are calculated based on point-wise mutual information (PMI) between a target word and the positive seeds and negative seeds, respectively [160, 104]. Similarly, PMI is used to build discrete emotion lexicon based on naturally annotated hashtags in twitter [103].

The second kind of approaches is based on label propagation methods which first build a word graph and then label propagation is performed to infer the affective values of unseen words from the seed words. Based on the different ways to construct the word graph, graph-based methods can be further divided into: (2A) Knowledge-based word graph such as building the graph based on ConceptNet [25], or WordNet [15, 129]. For example, a graph can be built based on the semantic relationship in WordNet and the label propagation is performed to infer the EPA values [2] and sentiment polarity[129]. A knowledge based graph is confined by the coverage of the knowledge base. (2B) Corpus based graph based on word cosine similarities which are calculated based on word-context co-occurrence statistics from a large corpus [162]; and (2C) Word embedding based graph based on the cosine similarity of available word embedding [184, 47]. For example, word embedding is used to compute the cosine similarity between words to build the word graph and PageRank algorithm is employed to infer the valence-arousal ratings of unseen words [184], random walk algorithm is used to infer the sentiment polarities [47].

The third kind of approaches represents a word as a vector and then map this vector to some sentiment value or categories based on a regression model or a classifier. This kind

of approaches mainly includes: (3A) representing words by manual defined features based on some knowledge base and performing linear regression on the features [168]; (3B) representing words as word embeddings obtained automatically and using a classifier [154] or linear regression [6] to obtain sentiment labels or scores; (3C) mapping word embedding into affective space through a transformation matrix that minimizes intra-group distance in each sentiment category and maximizes inter-group distance without considering the actual values of the seed words [126].

4.2 Regression Based Method

This work aims to make use of the semantic information encoded in word embedding to infer the affective meanings of words. This will help to build valuable lexical resources for emotion analysis using more comprehensive emotion models. The basic idea of the proposed approach is to use regression models to learn the affective meanings in each affective dimension. For a multi-dimensional emotion model having m dimensions, the objective is to learn m regression models that are suited for m affective dimensions separately. The proposed method is based on the assumption that word embedding has encoded the general semantic meaning into a dense vector and a certain dimension in word embedding contributes differently to different affective meanings. The proposed approach is a general learning method by using word embedding and regression through a set of seed words. This method is referred to as the **Regression on Word Embedding**, denoted as **RoWE**.

4.2.1 Distributed Word Embedding

The first step in RoWE is to build a high-quality feature representation for words using a vector space model (VSM), which represents a word through word embedding (also called word vector) [161]. As introduced in Section 2.3 on word representation, word

embedding can be obtained either by count-based methods or prediction-based methods. According to a comprehensive study done by [74], both methods obtain similar results. In other words, they are basically equivalent although some fine tuning may be needed. However, prediction-based methods have lower computation cost because it does not need to perform matrix factorization over a large co-occurrence matrix. Thus, in this work, only prediction-based methods are explored to obtain word embedding.

Prediction-based methods are based on neural networks and one of the most widely used models is Skip-Gram with Negative Sampling (SGNS) [94], which has been roughly introduced in Chapter 2. Here the details of SGNS are introduced. Given a corpus with vocabulary V and the extracted word-context pair set D, let $p(D = 1|w, c)$ be the probability that (w, c) comes from D and w, c are both in the vocabulary. Let $p(D = 0|w, c)$ be the probability that (w, c) does not come from D. The basic assumption of SGNS is that the conditional probability of $p(D = 1|w, c)$ should be high if c is the context of word w within an observation window and low otherwise.

Let $\vec{w}$ denote the embedding of w, and $\vec{c}$ denote the embedding of c. Then, $p(D = 1|w, c)$ is computed as:

$$p(D = 1|w, c) = \sigma(\vec{w} \cdot \vec{c}) = \frac{1}{1 + e^{-\vec{w} \cdot \vec{c}}}. \tag{4.1}$$

Both $\vec{w}$ and $\vec{c}$ are model parameters to be learned. The basic idea behind is that if word w and context c co-occur, their corresponding vectors should have close correlation, modeled by $\vec{w} \cdot \vec{c}$. The objective of negative sampling is to minimize the conditional probability:

$$p(D = 1|w, c_N) = \sigma(\vec{w} \cdot \vec{c_N}), \tag{4.2}$$

where c_N denotes a negative context of w, namely, context that does not co-occur with word w. The method randomly samples negative context c_N of w from V_W. Let P_D be the empirical unigram distribution where

$$P_D(c) = \frac{\#(c)}{|D|}. \tag{4.3}$$

Combining Formula 4.1 and 4.2, the objective for each word-context pair can be translated into maximizing:

$$log\sigma(\vec{w} \cdot \vec{c}) + k \cdot \mathbb{E}_{c_N \sim P_D}[log\sigma(-\vec{w} \cdot \vec{c_N})], \tag{4.4}$$

where k is the number of negative samples. For a given training corpus with a set of words V_W, the final objective function for the whole corpus is:

$$\begin{aligned} J = \sum_{w \in V_W} \sum_{c \in V_W} \#(w,c)(log\sigma(\vec{w} \cdot \vec{c}) \\ + k \cdot \mathbb{E}_{c_N \sim P_D}[log\sigma(-\vec{w} \cdot \vec{c_N})]). \end{aligned} \tag{4.5}$$

The obtained $\vec{w}$ and $\vec{c}$ are the word embedding and context embedding, respectively. The performance of the embedding can be affected by the hyperparameters, as discussed in [74]. Because finding the optimal word embedding is not our focus, the recommended settings from [74] is used for the SGNS model. Note that any learning model for word embedding can be used in this framework, including matrix factorization based word embedding [119], knowledge base with corpus based word embedding [141], and ensemble based word embedding [182], etc.

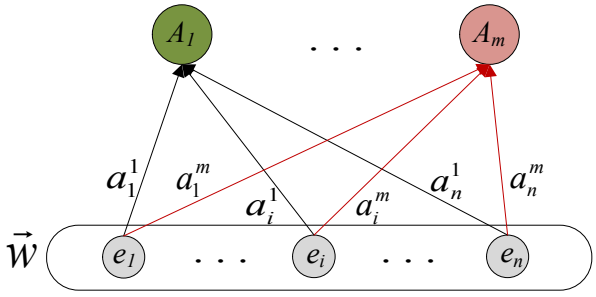


Figure 4.1: The proposed regression method for affective meaning prediction based on word embedding.

4.2.2 Regression for Affective Meaning Prediction

Figure 4.1 shows the regression based learning model from word embeddings to obtain

affective meanings of words. Given a set of seed words with affective values annotated in every dimension. In the training phase, each seed word s as a training sample has known word embedding $\vec{s}$ which is a vector of size n, and its affective meaning is defined in m dimensional space. A word embedding and an annotated affective value pair consists of one training sample. Given sufficient such pairs, a regression model can be learned for every affective dimension A_j where j is in the range of $[1...m]$. Based on the regression model in each dimension, the affective value of a new word can be predicted based on its word embedding. Consequently, an existing emotion lexicon can be extended automatically.

Given a seed, *s*, and its word embedding $\vec{w}^s = [e_1^s, e_2^s, ..., e_n^s]$, the following mapping function f_j for the jth affective dimension needs to be learned.

$$f_j(\vec{w}^s) = g_j(a_1^j e_1^s + a_2^j e_2^s + \cdots + a_n^j e_n^s), \tag{4.6}$$

where a_i^j is the weight of feature i, g_j is the mapping function. When f_j is a scalar value, g_j can be the identity function and this model becomes a typical linear regression model. When f_j takes categorical labels, g_j can be a logistic function and this model becomes a typical logistic regression model. f_j can be learned for any kind of affective meanings, including valence, arousal, dominance in the VAD model, evaluation, potency, activity in the EPA model, or a simple positive/negative label.

Let V denote the set of seed words. The objective function for regression learning in each affective dimension j is then defined as follows:

$$\min_{\vec{a}} \sum_{s \in V} ||f_j(\vec{w}^s) - y_j^s)||_2^2 + \alpha R(\vec{a}^j), \tag{4.7}$$

where $R(\vec{a}^j)$ is the regularization part on the weight vector $\vec{a}^j = [a_1^j, a_2^j, ..., a_n^j]$ and α is the regularization weight. When $\alpha = 0$, the model degrades to the ordinary least squares linear regression. When $\alpha \neq 0$ and $R(\vec{a}^j) = ||\vec{a}^j||_2^2$, the model degrades to the Ridge regression model. When $\alpha \neq 0$ and $R(\vec{a}^j) = ||\vec{a}^j||_1^1$, the model degrades to the Lasso regression model. Different regression models are evaluated in the experiments.

The proposed method can be trained on any emotion lexicon. After the model is learned, given the embedding of a new word, the affective meanings of the new word in m dimensions can be predicted using m regression models based on its word embedding. The size of the constructed lexicon should only be limited by the size of the available word embeddings, which is in principle unlimited because of plenty of available text corpora.

4.3 Experiments and Analysis

Six sets of experiments are reported in this section. The first set of experiments are performed to evaluate the proposed method in inferring affective meanings under different emotion models including sentiment and several multi-dimensional lexicons. To demonstrate the generality of the proposed method, predictions of other non-emotion dimensions are also tested, including concreteness-abstractness, perceptual strength in five senses of hearing, seeing, touching, tasting and smelling. The second set of experiments evaluate the complexity of the proposed method and the baseline methods. The third set of experiments evaluate the effects of seed words for different methods. The fourth set of experiments evaluate the effects of embedding dimension size. The fifth set of experiments look at the performance of different regression models and also examine the different embedding resources in terms their predictability on an existing lexicon. The last set of experiments evaluate the performance of sentiment lexicons obtained by RoWE on a downstream sentiment analysis task.

Experiment settings: In the following experiments for English dataset, the 300 dimensional word embedding is trained based on Wikipedia August 2016 dump[1] with 3.1 billion tokens and a vocabulary size of 204,981. Since the experiments also include a Chinese emotion lexicon, the 300 dimensional word embedding is trained based on Baidu Baike corpus with 1.8 billion tokens[2] after performing word segmentation using the HIT

[1] https://dumps.wikimedia.org/enwiki/latest/ Accessed May 17, 2017

[2] http://www.nlpcn.org/resource/list/2 Accessed May 17, 2017

LTP tool[3]. Both embeddings are trained using the SGNS model introduced in **Section 4.2.1**. The trained word embeddings are referred as **embedding lexicons.**

4.3.1 Inferring Affective Meanings

The first set of experiments is to examine the effectiveness of the proposed RoWE. The compared methods are listed below.

1. **PMI** [160]: This method learns the intensity value of a word measured by the pointwise mutual information (PMI) with the seed words.

2. **Web-GP** [162]: This web-based graph propagation method constructs a weighted graph using cosine similarity of a word and its context by a vector of co-occurrences. This method only keeps the 25 highest weighted edges for each node to reduce the effect of noise in web data. The iteration number is set to 5.

3. **QWN-PPV** [129]: This method extracts the set of words in WordNet that has polarity information [118]. Then a word graph is built based on relations in WordNet. PageRanking algorithm is used to obtain sentiment intensity of unseen words.

4. **DENSIFIER** [126]: This method learns an orthogonal transformation from the original embedding space to obtain task specific information in an ultradense space, such as the one-dimensional sentiment polarity space.

5. **SENTPROP** [47]: This method uses cosine similarity of word embedding as the edge weights to construct a word graph and uses the random walk algorithm to obtain the affective values.

6. **Wt-Graph** [184]: This method uses the cosine similarity of word embedding as the edge weight to construct a weighted word graph and uses the PageRank algorithm to obtain the affective values.

[3] www.ltp-cloud.com/ Accessed May 17, 2017

Other than QWN-PP which extracts sentiment words from WordNet, all the other methods need to use some seed words to infer the affective meanings of unseen words. For a fair comparison, all the methods in the evaluation use the same set of seed words, the same corpus, and the same test settings.

The gold emotion lexicons used for this set of experiments are chosen because they are manually annotated and are considered to have high quality. **Table 4.1** gives a summary of the gold lexicons where "Multi-dim" means multi-dimensional emotion lexicons. The table lists the lexicon names (**Lexicon**), their sizes (**Size**), the number of words in the lexicons which also appear in the respective word embedding lexicons (**Overlap #**), whether standard deviation of annotation is supplied or not (**std**), the emotion model (**emotion model**), and the annotation value range (**Range**). The first group gives three sentiment lexicons. **GI** [145] is a sentiment lexicon annotated with *positive, neutral, negative*. During prediction, class-mass normalization is used to give discrete labels as done in [47]. **VADER** [59] and **SemEval2015** [125] are sentiment lexicons annotated with intensity. **VADER** also contains standard deviation of the annotation. The second group gives five multi-dimensional emotion lexicons. **ANEW** [17] and **E-ANEW** [167] are manually annotated in the three dimensions of valence, arousal and dominance with values from 1 to 9. **E-ANEW** is an extended version of **ANEW** through crowdsourcing. **CVAW** [183] is the Chinese version of ANEW but annotated only on the two dimensions of valence and arousal. **EPA** [51] is annotated in the three dimensions of evaluation, potency and activity. **DAL** [169] (dictionary of affect in language) is annotated in the three dimensions of evaluation, activation and imagery (**EAI**), where the dimension of imagery measures how easily the word can bring an image to mind. The third group gives two lexicons used to measure perceptions and concreteness in psychology, respectively. These two are used here to test the generality of the different models to infer the other semantic meanings of a word. **Perceptual** [87, 88], used to measure sensory intensities of words, is annotated with a perceptual strength of a target word by feeling through five sensations. During an-

notation, each word is annotated through the question *"To what extent do you experience something being WORD"* (with *"WORD"* being the target word to be annotated). Underneath this question are five separate rating scales for each perceptual modality, labeled "by feeling through touch", "by hearing", "by seeing", "by smelling", and "by tasting". The participants are asked to rate the extent to which they would experience about the five senses, from 0 (not at all) to 5 (greatly) [87, 88]. **Concreteness** [19] is annotated on the degree of concreteness or abstractness of a word through crowdsourcing. Among those lexicons, only CVAW is Chinese and all the others are English.

Table 4.1: Summary of lexicons used in the experiments.

Type	Lexicon	Size	Overlap #	std	Emotion Model	Range
Sentiment	GI	3,626	2,942	N	Sentiment	$\{-1, 0, 1\}$
	SemEval2015	1,515	751	N	Sentiment	$[-1, 1]$
	VADER	7,502	3,124	Y	Sentiment	$[-4, 4]$
Multi-dim	ANEW	1,034	958	Y	VAD	$[1, 9]$
	E-ANEW	13,915	11,364	Y	VAD	$[1, 9]$
	CVAW (Chinese)	1,647	1,309	Y	VA	$[1, 9]$
	EPA	4,505	2,901	Y	EPA	$[-4, 4]$
	DAL	8,743	8,003	N	EAI	$[1, 3]$
Others	Perceptual	1,001	826	Y	Five senses	$[0, 5]$
	Concreteness	39,954	18,111	Y	Concreteness	$[1, 5]$

The respective overlap sets between the embedding lexicons and the gold lexicons are randomly split equally to form the training sets and the testing sets. Each experiment runs five times and the average result and standard deviation are reported. The standard deviation is used to measure the robustness of the methods. To satisfy the requirement of the bipolar scale of some baselines (PMI, Web-GP, DENSIFIER, SENTPROP), the affective scales are transformed to bipolar scales if needed. For example, ANEW, E-ANEW, and CVAW are mapped from $[1, 9]$ to $[-4, +4]$ linearly, DAL is mapped from $[1, 3]$ to $[-1, +1]$, Perceptual is mapped from $[0, 5]$ to $[-2.5, 2.5]$ and Concreteness is mapped from $[1, 5]$ to $[-2, 2]$. The final predicted values are mapped back to the annotation range.

For the regression model in RoWE, Ridge regression is used in the scikit-learn tool [4] with default parameters. The training data (seed word) size is set 50% of the total overlap set between the lexicons and the embedding vocabulary.

Evaluation metrics: For the GI lexicon, the classification is ternary. The AUC (area under a curve) and macro F-score (denoted as F1) are used as the evaluation metrics using the method given in [47] to transform the predicted scalar values to sentiment labels[5]. For all the other lexicons where the predictions are on continuous scales, the following evaluation metrics are used:

1. Root mean squared error (RMSE)
   $RMSE = \sqrt{\sum_{i=1}^{n}(A_i - P_i)^2/n}$,
2. Mean absolute error (MAE)
   $MAE = \frac{1}{n}\sum_{1}^{n}|A_i - P_i|$,
3. Mean absolute percentage error (MAPE)
   $MAPE = \frac{1}{n}\sum_{1}^{n}\frac{|A_i - P_i|}{A_i} \times 100\%$, and
4. Kendall rank correlation coefficient τ
   $\tau = \frac{C-D}{C+D}$,

where A_i is the gold standard value; P_i is the predicted value; n is the total number of the test samples; $\bar{A}$ and $\bar{P}$ are the average values of A and P; C is the number of concordant pairs; and D is the number of discordant pairs. The lower the values of RMSE, MAPE and MAE, and the higher the value of τ, the better the performance is. Note that the MAPE evaluation metric suffers from the so-called zero-division problem. In the experiment, the MAPE result is not reported if the gold value contains 0. So, for lexicons whose values

[4] http://scikit-learn.org/

[5] Though RoWE can directly predict discrete labels using logistic regression on word embedding, the baseline methods can only produce scalar values. To be consistent with the baselines, the scalar value is also predicted using a linear regression model.

contain zero (SemEval2015, EPA, DAL, Perceptual), the MAPE metric is not used because MAPE is sensitive to zero. In addition, for the lexicons with provided standard deviation on annotation, an additional evaluation metric on accuracy is defined as follows:

$$ac_{1\sigma} = \frac{1}{n}\sum_{i=1}^{n} g(\sigma_i - |A_i - P_i|), \tag{4.8}$$

where

$$g(x) = \begin{cases} 1: & x > 0, \\ 0: & otherwise. \end{cases}$$

σ_i is the annotated standard deviation; $\text{ac}_{1\sigma}$ indicates the percentage of correctly predicted samples within 1 standard deviation of the gold answers.

Table 4.2: Result on inferring affective meaning of sentiment on three sentiment lexicons.

	Method	PMI	Web-GP	QWN-PPV	DENSI	SENTP	Wt-Gp	RoWE	RI
GI	AUC	51.1(.68)	51.1(.94)	88.6(.39)	79.5(6.6)	72.3(5.7)	95.1(.25)	**96.2(.21)**	1.1
	F1	53.5(4.6)	48.2(1.2)	81.8(.67)	70.5(6.3)	64.5(5.4)	88.2(.48)	**89.4(.65)**	1.4
SemEval	RM	2.3(1.5)	.70(.05)	.47(.01)	4.8(1.6)	.56(.05)	.47(.01)	**.29(.00)**	62.1
	MA	2.1(1.6)	.54(.03)	.38(.01)	3.8(1.2)	.44(.05)	.37(.01)	**.23(.00)**	60.9
	τ	-.58(5.3)	-.47(4.1)	37.4(3.1)	-22.9(7.9)	16.3(6.7)	47.2(1.7)	**56.0(.85)**	26.2
VADER	RM	2.85(.15)	1.8(.01)	1.8(.01)	6.5(1.2)	1.8(.01)	1.7(.01)	**.96(.01)**	80.2
	MA	2.3(.16)	1.6(.00)	1.6(.01)	5.3(.91)	1.6(.01)	1.5(.01)	**.74(.01)**	102.7
	τ	.78(.56)	-1.2(2.2)	43.7(.95)	24.3(33.4)	27.8(5.0)	56.0(.65)	**62.0(.44)**	10.7
	$\text{ac}_{1\sigma}$	28.6(3.6)	20.8(.28)	22.6(.44)	9.8(1.6)	22.3(.51)	20.5(.20)	**63.6(.81)**	122.3

The performance evaluation results for the three of gold lexicons are shown from **Table 4.2** to **Table 4.9**. In all these tables, RM, MA, MP are used as the shorthand forms for RMSE, MAE, MAPE respectively. The best performing method is marked in bold. The second best performer is marked by an underline. The last column **RI** indicates the relative improvement of the best performer over the second best performer. RI is calculated as $RI = \frac{|p_{1st} - p_{2nd}|}{min\{p_{1st}, p_{2nd}\}}$, where p_{1st} is the best performance and p_{2nd} is the second performance.

Table 4.3: Result on inferring multi-dimensional affective meanings of VAD on ANEW.

	Method	PMI	Web-GP	QWN-PPV	DENSI	SENTP	Wt-Gp	RoWE	RI
Valence	RM	3.6	2.0	2.0	6.9	2.0	1.9	**1.2**	58.3
	MA	3.1	1.8	1.8	5.6	1.7	1.7	**0.91**	86.8
	MP	80.2	45.1	45.8	137.6	47.3	43.3	**22.0**	96.8
	τ	-0.25	-1.4	40.8	-4.4	17.8	52.9	**60.4**	14.2
	$ac_{1\sigma}$	31.8	49.0	49.0	19.2	52.7	54.2	**82.1**	51.5
Arousal	RM	2.5	1.1	-	5.5	1.1	1.0	**0.83**	20.5
	MA	2.3	0.91	-	4.4	0.93	0.84	**0.66**	27.3
	MP	49.2	18.3	-	86.8	20.8	17.6	**13.7**	28.5
	τ	-1.4	-0.53	-	-10.7	21.3	40.2	**43.5**	8.2
	$ac_{1\sigma}$	54.9	93.9	-	35.3	94.5	96.1	**97.9**	1.9
Dominance	RM	1.5	1.1	-	5.4	1.1	0.99	**0.75**	32.0
	MA	1.2	0.88	-	4.3	0.86	0.79	**0.59**	33.9
	MP	24.8	19.0	-	90.7	20.2	17.2	**12.6**	36.5
	τ	-0.92	0.72	-	3.6	11.8	46.3	**49.6**	7.1
	$ac_{1\sigma}$	81.7	94.8	-	29.7	94.7	97.0	**98.8**	1.9

Table 4.2 shows the result on the sentiment lexicons with standard deviations given in parenthesis. RoWE and Wt-GP are the best two performers while RoWE outperforms Wt-GP with large margins on SemEval and VADER lexicons as indicated by the respective RIs. For example, the average RI on SemEval and VADER over Wt-GP under the RM metric is 71.5%. The standard deviations indicate that RoWE has relatively smaller standard deviations under all evaluation metrics. In other words, RoWE is more robust and is less seed word sensitive. The average RI on SemEval and VADER is much larger (66.4% on average under different evaluation metrics) than on GI (1.3% on average under different evaluation metrics). This is because SemEval and VADER predict the precise sentiment value while GI only predicts the sentiment class, which maps to only a range of values in the continuous space. This also indicates that RoWE performs much better under stricter criteria. The much better result under $ac_{1\sigma}$ shows that the predicted values by RoWE are much closer to the gold answers than the baseline models.

Table 4.3 to **Table 4.7** are for multi-dimensional emotion lexicons. **Table 4.3** to **Ta-**

Table 4.4: Result on inferring multi-dimensional affective meanings of VAD on E-ANEW.

	Method	PMI	Web-GP	QWN-PPV	DENSI	SENTP	Wt-Gp	RoWE	RI
Valence	RM	2.0	1.3	1.3	5.2	1.3	1.2	**0.83**	44.6
	MA	1.7	1.0	1.0	4.1	0.99	0.96	**0.65**	47.7
	MP	40.2	23.3	23.2	88.5	24.8	22.7	**14.4**	57.6
	τ	-0.27	0.7	28.8	3.1	17.0	44.9	**53.4**	18.9
	$ac_{1\sigma}$	55.8	79.1	79.2	24.7	80.4	81.0	**93.4**	15.3
Arousal	RM	2.2	1.2	-	5.5	1.6	0.89	**0.74**	20.3
	MA	2.0	1.0	-	4.4	1.4	0.71	**0.58**	22.4
	MP	50.3	28.5	-	108.4	38.0	17.8	**14.5**	22.8
	τ	0.02	0.1	-	6.6	10.0	32.7	**38.1**	16.5
	$ac_{1\sigma}$	62.5	91.9	-	36.8	82.8	97.7	**99.1**	1.4
Dominance	RM	3.3	0.98	-	5.0	0.96	0.92	**0.71**	29.6
	MA	3.1	0.79	-	4.0	0.75	0.73	**0.56**	30.4
	MP	60.0	15.9	-	80.0	16.5	15.3	**11.5**	33.0
	τ	0.82	-0.26	-	-6.5	7.9	39.2	**44.2**	12.8
	$ac_{1\sigma}$	41.0	95.1	-	37.7	95.5	96.1	**98.9**	2.9

ble 4.5 are for the VAD lexicons of ANEW, E-ANEW, and CVAW, respectively. **Table 4.6** is for the EPA lexicon, and **Table 4.7** is for the DAL lexicon. QWN-PPV is included in ANEW, E-ANEW and DAL for the valence dimension. The results on the multi-dimensional lexicons from **Table 4.3** to **Table 4.7** show very similar performance compared to the analysis on the sentiment lexicons. RoWE performs much better than the second best performer, Wt-GP, on every affective dimension under all the evaluation metrics. Comparing between different dimensions shows that the relative improvement on valence dimension is much better than on the other dimensions. For example, in **Table 4.3**, the average relative improvement on valence is 61.5% while the average relative improvements on arousal and dominance are 17.3%, 22.3% respectively. Further analysis on the gold answers shows that the annotation standard deviations of valence, arousal, dominance are 1.65, 2.37, 2.06, respectively. A smaller standard deviation indicates better consistence between annotators and more accurate gold values. The Spearman rank-order correlation coefficient between the relative improvements and the annotation standard deviation is -1.

Table 4.5: Result on inferring multi-dimensional affective meanings of VA on CVAW.

	Method	PMI	Web-GP	DENSI	SENTP	Wt-Gp	**RoWE**	RI
Valence	RM	2.2	1.9	8.4	1.9	1.7	**0.83**	104.8
	MA	1.9	1.7	7.0	1.7	1.5	**0.64**	134.4
	MP	48.9	45.1	192.4	49.3	38.5	**16.7**	130.5
	τ	-3.0	0.23	25.1	43.6	59.9	**65.4**	9.2
	$ac_{1\sigma}$	15.4	12.5	9.4	12.4	12.8	**58.2**	277.9
Arousal	RM	1.5	1.4	6.3	1.2	1.2	**0.87**	37.9
	MA	1.2	1.1	5.2	0.96	0.95	**0.69**	37.7
	MP	22.4	19.6	95.5	18.6	19.0	**13.5**	37.8
	τ	-1.5	0.42	-9.4	12.6	39.2	**48.9**	24.7
	$ac_{1\sigma}$	57.2	59.7	15.0	64.8	66.0	**80.3**	21.7

Table 4.6: Result on inferring multi-dimensional affective meanings on EPA.

	Method	PMI	Web-GP	QWN-PPV	DENSI	SENTP	Wt-Gp	RoWE	RI
E	RM	2.7	1.4	1.4	4.5	1.3	1.3	**0.88**	47.7
	MA	2.4	1.2	1.2	3.5	1.1	1.0	**0.68**	47.1
	τ	-0.77	0.97	31.5	10.3	20.4	42.4	**51.3**	21.0
P	RM	2.2	0.87	-	4.5	0.74	0.75	**0.6**	23.3
	MA	2.0	0.67	-	3.6	0.58	0.59	**0.47**	23.4
	τ	-0.02	-0.69	-	3.7	0.9	34.4	**39.3**	14.2
A	RM	3.3	1.0	-	5.1	0.86	0.86	**0.7**	22.9
	MA	3.2	0.85	-	4.1	0.68	0.67	**0.54**	24.1
	τ	0.24	0.16	-	3.3	7.4	32.7	**40.2**	22.9

This indicates that the more accurate of the gold answers are, the better performance of RoWE compared to the baselines. For the E-ANEW lexicon, which is annotated through crowdsourcing, the mean absolute errors (MAE) of RoWE are 0.65, 0.58, 0.56 on valence, arousal, dominance, respectively. This means that the predicted values are quite close to the manually annotated values. On the $ac_{1\sigma}$ metric, RoWE achieves 93.4%, 99.1%, 99.0% on valence, arousal, dominance, respectively. This means that almost all the predicted values are within one standard deviation of the manually annotated mean value.

Table 4.8 and **Table 4.9** show the performance of different methods for perceptual

Table 4.7: Result on inferring multi-dimensional affective meanings of EAI on DAL.

Method		PMI	Web-GP	QWN-PPV	DENSI	SENTP	Wt-Gp	RoWE	RI
Evaluation	RM	2.3	0.48	0.44	4.8	0.75	0.43	**0.34**	26.5
	MA	2.3	0.38	0.34	3.8	0.66	0.33	**0.27**	22.2
	MP	130.7	23.2	21.4	215.2	41.8	20.6	**15.3**	34.6
	τ	0.57	-0.11	19.1	0.82	8.9	36.5	**40.8**	11.8
Activity	RM	2.0	0.45	-	4.6	0.71	0.39	**0.33**	18.2
	MA	1.9	0.36	-	3.7	0.63	0.31	**0.26**	19.2
	MP	108.1	21.7	-	204.7	39.0	18.9	**14.9**	26.8
	τ	0.19	0.15	-	8.6	3.3	28.7	**34.9**	21.6
Imagery	RM	2.5	0.64	-	5.0	0.81	0.6	**0.45**	33.3
	MA	2.3	0.53	-	4.0	0.68	0.5	**0.36**	38.9
	MP	132.0	32.2	-	232.8	46.9	31.3	**20.9**	49.8
	τ	1.7	0.75	-	-2.2	20.9	43.2	**50.1**	16.0

and concreteness lexicons, respectively. Results on the two lexicons also indicate similar conclusions as those on the emotion lexicons. For example, in **Table 4.9**, the average RI under different evaluation metrics over the second best performer is 60.9%. This whole set of experiments show that word embedding is very effective in predicting semantic meanings not only for affective aspect, but also for other meaning dimensions, as long as some seed words with the meanings defined quantitatively.

In conclusion, the proposed RoWE method achieves the best result on all the lexicons under all the evaluation metrics, which validates the assumption that word embeddings do encode semantic information and the regression model can effectively decode the affective meanings and other semantic meanings from the embeddings by assigning different weights to different dimensions in the embedding.

4.3.2 Case Study

Figure 4.2 shows a visualized weight values of $\vec{a}$ on the first ten dimensions of word embedding to the three affective dimensions on ANEW lexicon. Note that the weights for the three affective dimensions can be quite different. For example, for the first dimension

Table 4.8: Result on inferring five-sense meanings.

Method		PMI	Web-GP	DENSIFIER	SENTPROP	Wt-Graph	RoWE	RI
Hearing	RM	3.8	1.5	6.9	1.7	1.2	**0.91**	31.9
	MA	3.5	1.3	5.6	1.5	1.0	**0.73**	37.0
	τ	-0.11	0.01	-1.0	34.5	48.0	**50.8**	5.8
	$ac_{1\sigma}$	9.4	54.3	17.7	47.7	63.3	**76.6**	21.0
Tasting	RM	1.7	2.2	8.8	2.7	1.1	**0.73**	50.7
	MA	1.3	2.0	6.9	2.6	0.82	**0.52**	57.7
	τ	0.92	-3.8	14.4	17.2	35.7	**40.0**	12.0
	$ac_{1\sigma}$	40.8	17.1	6.8	12.7	45.0	**61.4**	36.4
Touching	RM	1.9	1.5	6.1	1.6	1.3	**0.96**	35.4
	MA	1.7	1.3	4.9	1.4	1.1	**0.8**	37.5
	τ	1.2	0.89	-3.3	12.6	39.9	**49.0**	22.8
	$ac_{1\sigma}$	47.2	55.9	18.6	53.3	61.0	**75.9**	24.4
Smelling	RM	3.0	2.0	6.2	2.5	1.0	**0.77**	29.9
	MA	2.8	1.9	4.8	2.4	0.83	**0.58**	43.1
	τ	-2.3	0.44	5.9	8.9	29.3	**37.9**	29.4
	$ac_{1\sigma}$	16.9	23.4	12.1	16.6	51.8	**66.6**	28.6
Seeing	RM	1.5	1.5	5.5	1.1	0.87	**0.71**	22.5
	MA	1.2	1.3	4.4	0.96	0.69	**0.56**	23.2
	τ	0.3	-0.13	-7.4	2.3	37.4	**41.5**	11.0
	$ac_{1\sigma}$	56.2	47.3	17.8	60.1	78.8	**85.9**	9.0

in embedding, its corresponding affective weights are 1.11, -1.05, and 0.63, respectively.

Table 4.10 lists some example words in the ANEW lexicon that are close in embedding space but not close in the valence dimension. In the table, the **Word** column is the target word, the **G val** column is the gold valence value, **P val** is the predicted valence value, and the last column is the top 5 nearest words in embedding space based on cosine similarity. The value in the parenthesis is the predicted valence value. The words in bold are examples that are close in the embedding space but not close in the valence dimension. For example, for the word *cold*, its nearest word is *warm* while their predicted valence value are 4.16 and 7.09, respectively. This validates that the proposed method can distinguish the affective meanings through assigning different weights to the features in the embedding space.

Table 4.9: Result on inferring concreteness.

	Method	PMI	Web-GP	DENSIFIER	SENTPROP	Wt-Graph	RoWE	RI
Concreteness	RM	2.3	1.0	5.9	1.0	0.97	**0.56**	73.2
	MA	2.1	0.89	4.8	0.89	0.84	**0.44**	90.9
	MP	71.3	31.0	178.3	35.4	30.8	**16.0**	92.5
	τ	-0.46	-0.45	18.5	34.9	56.2	**64.4**	14.6
	$ac_{1\sigma}$	32.9	66.8	15.1	63.9	67.9	**90.6**	33.4

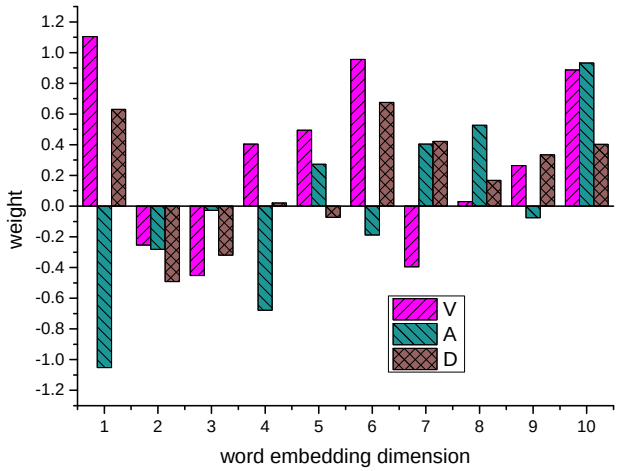


Figure 4.2: The learned weights of different affective meanings for the ANEW lexicon.

Table 4.11 lists some negative examples whose predicted valence value (**P val**) are not within 2 standard deviation of the gold values of (**G val**). The words in bold are dissimilar in the affective space with the target words. Detailed analysis reveals two possible reasons for wrong prediction. First, a word may have multiple senses which word embedding representation cannot distinguish because it is an inclusive mechanism to encompass the different senses in one presentation. Take the word *sad* in **Table 4.11** as an example, the top 7 closest words include *novi, pazar, etc* because there is a city called *Novi Sad*, which is the second largest city of Serbia. Similar situations occur for the word *engaged*. Secondly, a word normally has similar context to its antonyms which means they also have similar representation although the affective meaning are likely to be quite different. Taking the word *sad* as an example again, its closest word contain *happy*, which has opposite valence.

Table 4.10: Example words close in embedding space but not close in affective space. Words in bold are dissimilar in affective space with the target words.

Word	**G val**	**P val**	**Top 5 nearest words in embedding space**
good	7.47	6.45	decent(5.94), **bad(3.34)**, excellent(7.35), **poor(3.32)**, commendable(7.19)
heaven	7.3	6.80	heavens(6.33), heavenly(6.80), **hell(4.74)**, god(6.54), afterlife(5.63)
clouds	6.18	5.66	cloud(5.00), mist(5.00), droplets(4.85), **dust(4.27)**, **overcast(4.54)**
cold	4.02	4.16	**warm(7.09)**, winters(5.27), colder(4.94), cool(6.34), freezing(4.24)
displeased	2.79	3.64	angered(3.34), unhappy(3.43), incensed(3.37), **pleased(6.40)**, apprehensive(3.79)

Table 4.11: Negative examples whose predicted values are not within 2 standard deviations of the gold value.

Word	**G val**	**P val**	**Top 7 nearest words in embedding space**
humor	8.56	6.63	humour, **irreverent**, **satire**, **irony**, wry, humorous, **sarcasm**
engaged	8.00	5.05	engaging, engage, engages, **involved**, **participated**, resumed, **commenced**
rescue	7.70	4.53	rescues, rescuing, rescued, rescuers, **firefighting**, salvage, ambulance
sad	1.61	4.79	**novi**, **pazar**, **kragujevac**, **subotica**, pathetic, **happy**, melancholic

Some words used as training data also have the above two problems intrinsically, which can have adverse effects on the learned regression model.

4.3.3 Computation Efficiency Analysis

The second set of experiments examines the run time efficiency of different methods presented in Section 4.3.1. This experiment visually observes the difference in computing time by varying the data size from 1,000 to 11,000 using the E-ANEW lexicon. The size of the seed words is 300. The remaining collection is used as test data. The hardware platform is a desktop computer with processor of Intel (R) Xeon (R) CPU E5-1620 and 64G RAM. During the running of each method, all the other programs are closed. The result is shown in **Figure 4.3**.[6] Web-GP is not listed because its running time is too high, ranging from about 23,900 to 38,000 (in micro seconds). The figure shows that RoWE requires the least running time. When the data size increase from 1,000 to 11,000, the running time of RoWE changes from 11 to 116 which translates to a linear increase of 10.5 times.

Since running time may also be affected by implementation efficiency and environment, complexity in terms of Big O analysis is listed in **Table 4.12** for theoretic analysis. In this table, N is the data sample size, d is the embedding dimension and k is the number of nearest neighbors used in Web-GP and SENTPROP. d and k are fixed and thus considered as constants. The second column indicates that the asymptotic complexities of PMI, Web-GP, Wt-Graph and SENTPROP grow quadratically with the data size, whereas the complexities of DENSIFIER and RoWE grows linearly with the data size. The third column shows the complexity with constant coefficients d and k. Even though d and k do not have a role in Big O analysis, as shown in the second column, they do affect the efficiency of the implementations especially when data samples have limited size. In conclusion, RoWE has the same complexity as DENSIFIER ($O(N)$), and lower complexity

[6] The y axis is broken at 5000 to 6000 to make the figure more readable. The numbers in parenthesis are the running time.

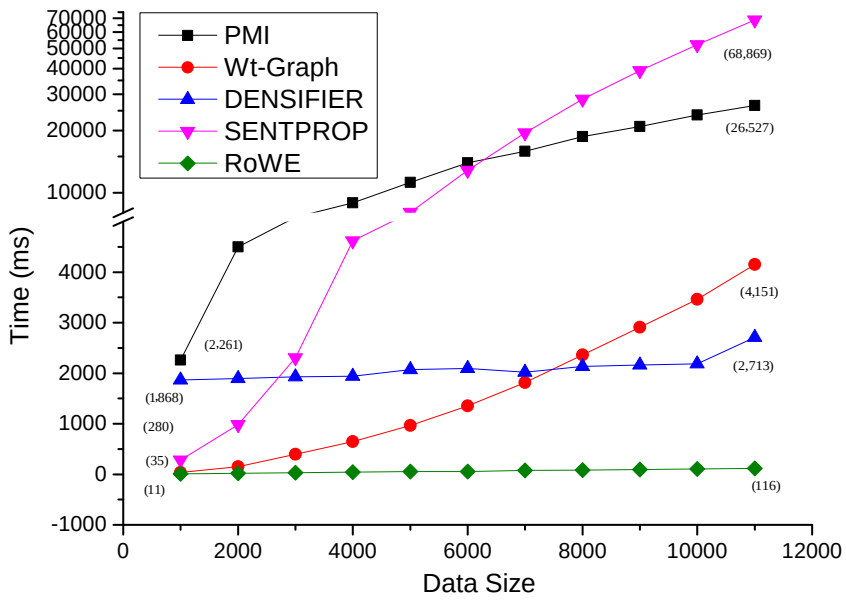


Figure 4.3: The running time of different methods under different data size.

than all the other methods. With consideration of coefficients k and d, RoWE is the most efficient method theoretically.

Table 4.12: Complexity of different methods.

Method	**Asymptotic Complexity**	**Complexity with coefficient**
PMI	$O(N^2)$	$O(N^2)$
Web-GP	$O(N^2)$	$O(N^2kd)$
Wt-Graph	$O(N^2)$	$O(N^2d)$
DENSIFIER	$O(N)$	$O(Nd^3)$
SENTPROP	$O(N^2)$	$O(N^2kd)$
RoWE	$O(N)$	$O(Nd^2)$

4.3.4 The Effects of Seed Words

In the third experiment, the effects of seed word size are explored using the ANEW lexicon. The size of the seed words ranges from 10 to 800 with 30 as the step size and the remaining as the test data. The result on the valence dimension in terms of $\text{ac}_{1\sigma}$ is shown in **Figure 4.4**, which indicates that Web-GP, SENTPROP, and Wt-Graph achieve almost a

similar result and they are stable without much room to improve when more seed words are added. PMI and DNSIFIER, however, are not stable. RoWE on the other hand, has much better performance. Even with a small set of seed words (such as 100, which can be obtained easily through manual annotation), RoWE still achieves a much better result. Also, when more seed words are used, the performance continues to improve.

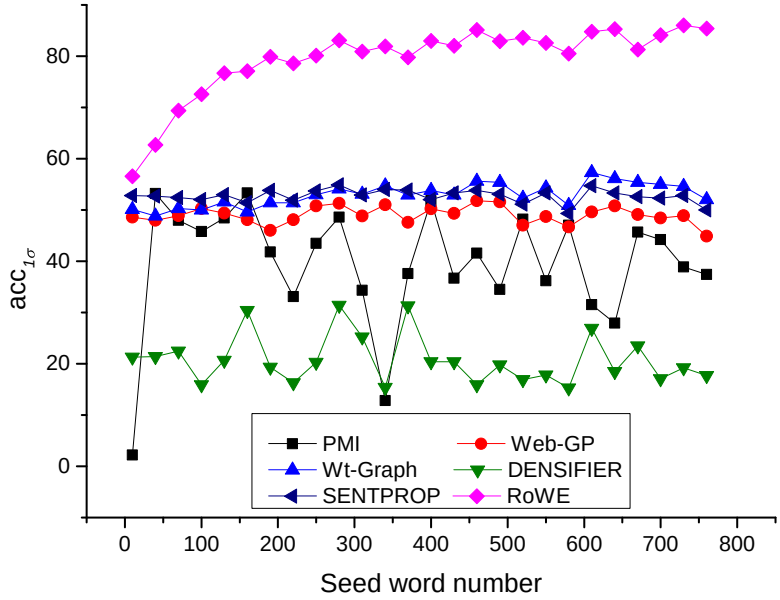


Figure 4.4: The effects of seed word size.

4.3.5 The Effects of Word Embedding Dimension

In the fourth experiment, the effects of embedding dimension size are explored. Word embeddings are trained on the Wikipedia corpus with different dimension size using the SGNS model and RMSE is used as the indicator on the VADER lexicon and the E-ANEW lexicon. The result is shown in **Figure 4.5**. Note that as the dimension increases from 50 to 300, the performance improves steadily. However, between 300 to 500, the curve becomes quite flat. Generally speaking, larger dimensions do bring better performance, but it would require more resources and computation power. To balance the performance and computation cost, the dimension is suggested to be set between 300 to 400.

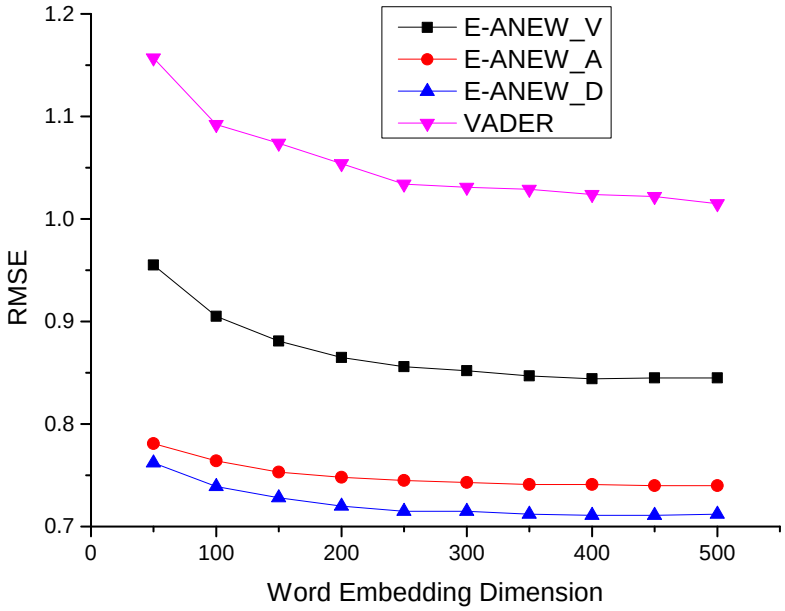


Figure 4.5: The effects of word embedding dimension.

4.3.6 The Effects of Regression Models and Word Embeddings

Previous experiments use the Ridge regression model and the word embedding trained using the SGNS model. RoWE, in principle, has no restriction on the regression model nor the word embedding learning method. In practice, however, different regression models and the actual embedding learning models may affect the overall performance. In this section, the effects of the regression models and word embedding models are investigated.

As summarized in Section 4.2, typical regression models include linear regression, Ridge regression, BayesianRidge regression, ElasticNet regression, Lasso regression, as well as Support Vector Regression with linear kernel (SVM-Linear), Support Vector Regression with non-linear Gaussian kernel (SVM-RBF). The performance of these models is evaluated in terms of $ac_{1\sigma}$, using the one-dimensional VADER lexicon. The size of the seed words changes from 10 to 600 with 30 as the step size and the remaining as the test data. All the models are implemented using the scikit-learn[7] tool with default parame-

[7] scikit-learn.org/ Accessed May 17, 2017

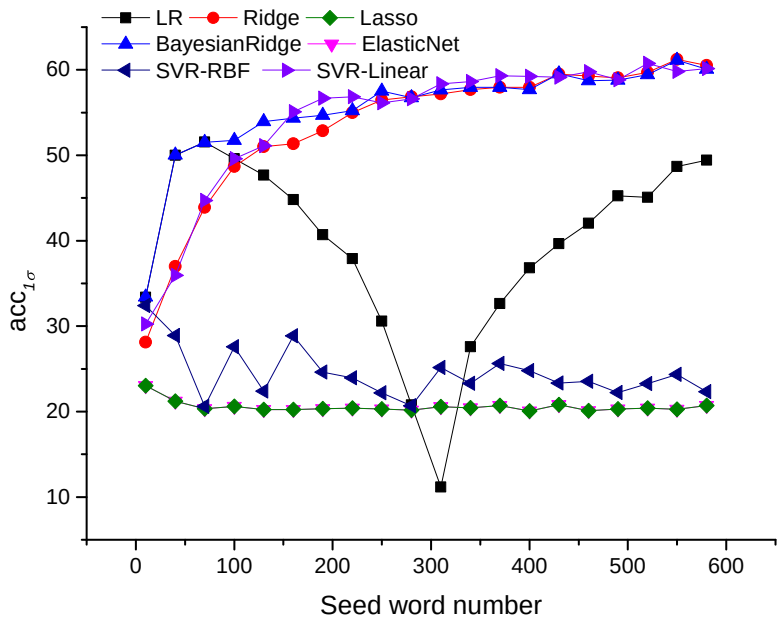


Figure 4.6: The performance of different regression models on the VADER lexicon.

ters. The result is shown in **Figure 4.6**. Note that the SVR-Linear, Ridge and Bayesian Ridge achieve similar and much better result than the other regression models. This is because Ridge regression and SVR-Linear use norm 2 regularization on the weights to avoid overfitting. The linear regression model has a typical U shape because of overfitting without regularization on the weight coefficients. SVR-Linear performs much better than SVR-RBF. This indicates that linear models are more suitable than non-linear models for inferring affective meanings from word embedding. Similar results are obtained under other evaluation metrics and other emotion lexicons. Thus, the suggestion is to use SVM-Linear, Ridge or Bayesian Ridge regression models in RoWE framework.

Next, different embedding resources are evaluated. In addition to the Wikipedia word embedding lexicon, denoted as **wikiEmb** with size 204,981, as explained in **Section 4.3.3**, the following public available embeddings that are obtained from different learning methods are compared:

1. Google embedding (**GoogleEmb**) [94]: It is trained using the SGNS model as in-

troduced in **Section 4.2.1** from a news corpus of 10 billion tokens.[8] The embedding vocabulary size is 3,000,000.

2. Glove 840B (**Glove**) [119]: It is based on weighted matrix factorization on the co-occurrence matrix built from a corpus consisting of 840 billion tokens.[9] The embedding vocabulary size is 2,196,017.

3. Meta-Embedding (**MetaEmb**) [182]: This method ensembles different embedding sources to obtain the final meta-embedding.[10] The size is 2,746,092

4. ConceptNet Vector Ensemble (**CNVE**) [141]: This method combines word2vec, Glove with structured knowledge from ConceptNet [143] and PPDB [40].[11] The size is 426,572.

5. MVLSA (**MVEmb**) [122]: This method learns word embedding from multiple sources including text corpus, dependency relation, morphology, monolingual corpus, knowledge base from FramNet based on generalized canonical correlation analysis.[12] The size is 361,082.

6. Paragram Embedding (**ParaEmb**) [171]: This method learns word embeddings based on the paraphrase constraint from PPDB.[13] The size is 1,703,756.

The experiment uses this seven word embedding lexicons to predict the affective meanings of 1,079 words common in all the selected embedding resources against the VADER emotion lexicon. Among the 1,079 words, 50% are randomly selected as seed words and the other 50% as test words. Each experiment is run 5 times and the average performance

[8] https://code.google.com/archive/p/word2vec/ Accessed May 17, 2017

[9] http://nlp.stanford.edu/projects/glove/ Accessed May 17, 2017

[10] http://cistern.cis.lmu.de/meta-emb/ Accessed May 17, 2017

[11] https://github.com/commonsense/conceptnet-numberbatch Accessed May 17, 2017

[12] `http://cs.jhu.edu/~prastog3/mvlsa/` Accessed May 17, 2017

[13] `http://ttic.uchicago.edu/~wieting/` Accessed May 17, 2017

within one standard deviation in the parenthesis is reported in **Table 4.13**. Note that the knowledge based CVNE achieves the best result under all the evaluation metrics. Other than MVEmb, which seems to be low in performance, all the other embeddings have comparable performance. This indicates that distilling knowledge base into embedding can improve the quality of word embedding. GoogleEmb performs slightly better than wikiEmb because GoogleEmb uses a much larger training corpus. Detailed discussion on the quality of embedding methods can be found in [74]. Even though CVNE has the best performance in this experiment, it only indicates the usefulness of adding knowledge base information to a non-supervised training method. It does not by any means guarantee that CVNE is the best performer on a downstream task because lexicon size is limited by the coverage of the knowledge base.

Table 4.13: Evaluation of different embeddings on VADER lexicon using RoWE.

Method	**RMSE**	**MAE**	τ	$\mathbf{ac}_{1\sigma}$
wikiEmb	1.2(.02)	.96(.01)	49.9(1.1)	53.6(1.0)
GoogleEmb	1.1(.01)	.86(.01)	55.4(1.0)	57.6(1.5)
Glove	1.0(.02)	.80(.02)	59.4(1.2)	61.7(1.5)
MetaEmb	1.1(.03)	.86(.02)	56.4(1.3)	57.8(1.4)
CVNE	**.88(.01)**	**.69(.01)**	**66.0(.95)**	**67.3(1.2)**
MVEmb	1.3(.02)	1.0(.02)	42.4(1.0)	50.7(.31)
ParaEmb	1.0(.02)	.80(.02)	59.6(1.4)	60.8(1.4)

Table 4.14 shows the example words with the top 5 largest and top 5 smallest predicted values in each affective dimension under different emotion models using CVNE embedding. Note that this list not only contains words, but also phrases because CVNE also includes embeddings for some frequently used phrases. Since RoWE has no restriction on the unit size for learning the affective meanings, phrase prediction is not a problem in general as long as phrase embeddings are given. All the learned top ranked words are quite reasonable. As sentiment indicators, ANEW-v and EPA-e have the same word *giving gift*. Several words do get listed in different lexicons such as *giving gift, make happy*.

Table 4.14: Example words with top 5 largest and smallest predicted affective values based on CVNE embedding.

Examples words of top 5 largest predicted affective values	
VADER	giving gift, making happy, excellentness, life of party, winning baseball game
ANEW-v	giving gift, making happy, make happy, reading books, positive attitude
ANEW-a	insanity, gun, sex, rampage, tornado
ANEW-d	paradise, win, positive attitude, incredible, self
EPA-e	giving gift, heaven, make happy, making happy, positive attitude
EPA-p	god, ceo, christ, herculean strength, pope
EPA-a	raver, riot, gunfight, fighter, nightclub
DAL-e	giving gift, making happy, make happy, showing love, enjoying day
DAL-a	dangerous activity, climbing mountain, playing snooker, winning game, playing cricket
DAL-i	neighbor's house, non powered device, own home, opaque thing, single user device
Concreteness	non powered device, opaque thing, power shovel excavator, non agentive artifact, single user device
Examples words of top 5 smallest predicted affective values	
VADER	hell with ,unpleasant person ,hagridden ,abusive language ,hagride
ANEW-v	stabbing to death , life threatening condition , poor devil , crybully , abusive language
ANEW-a	soothing , librarian , dull , calm , grain
ANEW-d	uncontrollable , earthquake , lobotomy , alzheimers , dementia
EPA-e	hell , murder , rape , unpleasant person , rapist
EPA-p	coward , weakling , high and dry , slave , powerless
EPA-a	glum , cemetery , funeral , mummy , graveyard
DAL-e	mommick , unpleasant person , plague , plaguer , nidder
DAL-a	scar , shadows , elementary , supplement , oxgang
DAL-i	that degree , risibility , in such way , inhere , in this
Concreteness	more equal, confessedly, hypostatize, neuter substantive, istically

Interestingly, on Concreteness, the last word *istically* is an adverb suffix, which is indeed quite abstract although quite unexpected. More samples of extended multi-dimensional emotion lexicons can be found at **Appendix** B.

4.3.7 Downstream Task for Sentiment Classification

The sixth experiment evaluates the effectiveness of RoWE through a downstream sentiment analysis task. This experiment examines the effectiveness of the lexicons obtained from RoWE compared to the baseline lexicons obtained from other methods including both manual ones and automatically obtained ones. Eight sentiment corpora used in this experiment are listed in **Table 4.15**, which are annotated with positive or negative labels. The eleven baseline lexicons listed in **Table 4.16** are openly available for access. The lexicons are sorted according to their size. Other than the three emotion lexicons, ANEW, VADER and E-ANEW, which are obtained either manually or through crowdsourcing, all the others are obtained automatically.

The setup of the experiment is first to use RoWE to extend the VADER sentiment lexicon using different word embedding lexicons introduced in **Section 4.3.6**. RoWE is trained using the intersection of the VADER lexicon and the respective word embeddings. The size for each of the extended lexicons is different depending on the vocabulary of the word embeddings. VADER is chosen because it is the largest manually-annotated sentiment lexicon in the bipolar format currently. For a fair comparison, only one downstream sentiment classifier is used for all different emotion lexicons. The sentiment classification method is used from [59] which is a simple heuristic rule-based method using a list of lexical features (along with their associated sentiment intensity measures) defined manually. No machine learning methods are used to avoid mixing other factors to introduce unknown variants to the evaluated lexicons. The selected method is very suited to measure the quality of the evaluated sentiment lexicons. In the sentiment analysis task, F-score is used as the evaluation metric.

Table 4.15: Statistics of sentiment corpora

Corpus	num	pos num	vocab	avg words	Description
SST[139]	1,821	909	7,576	19.2	movie review
sem[108]	3,583	2,570	18,965	19.8	SenEval 2013
aR[59]	3,708	2,128	8,306	16.5	Amazon review
cr [57]	3,771	2,405	5,712	20.1	customer review
nyt[59]	5,190	2,204	20,929	17.5	News
mpqa [170]	10,603	3,311	6,298	3.1	news
mR[59]	10,605	5,242	29,864	18.9	movie review
mr [113]	10,662	5,331	21,425	21.0	movie review

Table 4.16: Statistics of baseline sentiment lexicons

Lexicon	size	Description
ANEW[17]	1,034	manual annotation
VADER[59]	7,502	crowdsourcing annotation
E-ANEW[167]	13,915	crowdsourcing annotation
SenticNet4[24]	50,000	propagation on ConceptNet
HashtagSenti[191]	54,129	statistics based on hashtag
senti140[191]	62,468	statistics based on emoticon
QWN-PPV[129]	81,248	propagation on WordNet
SentiWordNet3[7]	89,631	automatic based on WordNet
SentiWords[41]	147,305	ensemble on SentiWordNet
NNlexicon[156]	184,579	neural network prediciton
Tang[154]	347,446	representation learning

Table 4.17 shows the evaluation result and the best results are marked in bold. In this table, each row represents a lexicon and each column is one sentiment corpus. The first part lists all the baseline lexicons and the second part lists the extended lexicons by RoWE based on different embeddings and the size of each obtained lexicon is included in parenthesis. In general, the embedding based lexicons perform better than the baseline lexicons. The ParaEmb lexicon, in particular, achieves the best result on all the sentiment corpora. In the baseline lexicons, SentiWords performs the best. Note that in both the baseline lexicons and lexicons obtained from RoWE, lexicon size is not the determiner for the best performance. Among the baseline lexicons, the best performer, SentiWords has only about

Table 4.17: Result on downstream sentiment analysis task

Lexicon(size in M)	sem	mR	aR	nyt	cr	mpqa	mr	SST
ANEW	0.71	0.56	0.55	0.49	0.62	0.27	0.54	0.57
VADER	0.83	0.66	0.71	0.57	0.78	0.63	0.66	0.70
E-ANEW	**0.85**	0.68	**0.74**	0.63	**0.79**	0.58	0.68	0.70
SenticNet4	0.79	0.66	0.69	0.59	0.74	0.57	0.66	0.68
HashtagSenti	0.81	0.62	0.66	0.53	0.71	0.41	0.62	0.66
senti140	0.82	0.68	0.65	0.60	0.68	0.55	0.68	0.70
QWN-PPV	0.76	0.63	0.69	0.57	0.74	0.45	0.63	0.66
SentiWordNet3	0.65	0.56	0.56	0.49	0.62	0.43	0.56	0.60
SentiWords	**0.85**	0.68	**0.74**	0.63	**0.79**	0.60	0.68	0.71
NNlexicon	0.77	0.64	0.68	0.53	0.73	0.55	0.64	0.67
Tang	0.83	0.63	0.63	0.53	0.66	0.54	0.63	0.68
wikiEmb(0.2M)	0.84	0.68	**0.74**	0.62	0.78	0.66	0.68	0.69
GoogleEmb(3M)	**0.85**	0.68	**0.74**	0.63	0.78	0.68	**0.69**	0.70
Glove(2M)	**0.85**	**0.69**	**0.74**	**0.65**	**0.79**	0.69	**0.69**	0.71
CVNE(0.4M)	**0.85**	**0.69**	**0.74**	0.63	0.78	0.68	**0.69**	0.70
MetaEmb(2.7M)	0.73	0.47	0.49	0.43	0.48	0.04	0.49	0.47
MVEmb(0.36M)	**0.85**	0.68	**0.74**	0.62	0.78	0.68	0.68	0.69
ParaEmb(1.7M)	**0.85**	**0.69**	**0.74**	**0.65**	**0.79**	**0.70**	**0.69**	**0.72**

147K sentiment words whereas NNLexicon and Tang have about 184K and 347K respectively. The best performer ParaEmb is also not the largest in lexicon size. In fact, CVNE which is only 0.4M in size has very good performance. One likely explanation of their good performance is that ParaEmb include knowledge base information which enriches the lexicon semantically. Note that MetaEmb performs much worse than other embedding based lexicons. Further analysis indicates that although the size of MetaEmb is large, the overlap size of MetaEmb with the sentiment corpora vocabulary is quite small. For example, there are only 512 overlapping seeds out of 6,298 (10%) in the mpqa corpus compared to 6,193 of ParaEmb. Also, most of the words in MetaEmb are informal strings, such as *rates.download, now!download*. The general conclusion is that (1) the larger overlapping is generally good, but again it is not the determining factor; and (2) the high-quality word embedding also helps even if its size is not large (as shown by CVNE).

4.4 Chapter Summary

In this paper, a regression based method is proposed to automatically infer the affective meanings of words from word embedding. Word embedding not only carries general semantic meanings but also meanings in some specific space, such as affective meanings. This framework first learns word embeddings through unsupervised way and then treats each word embedding as feature representation to train a Ridge regression model based on a small set of seed words. The proposed framework can infer different kinds of affective meanings in multi-dimensional models. A whole set of evaluations shows that: 1) The proposed RoWE achieves the state-of-the-art performance, outperforming all the baseline methods on several emotion lexicons in affective space and lexicons in other semantic space; 2) The proposed RoWE is rating scale insensitive, which means that the method does not require the rating range to be bipolar and there is no need to transform unipolar ratings to bipolar ratings; 3) The proposed RoWE is computationally more efficient than the baseline methods, especially compared to propagation based methods; 4) One experiment using the built sentiment lexicon on the downstream sentiment analysis task shows that lexicons based on word embedding perform better than previously available sentiment lexicons; 5) Comparing between different word embeddings, the promising result of CVNE which incorporates knowledge base gives a future research direction to obtain better word embedding with an incorporation of knowledge base information; and 6) The proposed RoWE provides valuable additional emotion lexicon resources for dimension-based EA. The extended lexicons with about million of words using different emotion models are available.[14] The dimensional emotion lexicons are one kind vector representations of words. Can they be used in the composition models to infer the emotion or sentiment of larger text units? The next chapter will investigate the validity of employing dimensional emotion lexicons under different composition models for emotion analysis.

[14] `https://yunfeilongpoly.github.io/Team_resource.html` Accessed December 06, 2017.

Chapter 5

Phrase Level Emotion Analysis

One premise of machine learning based emotion analysis is to obtain feature representations of a target text. Then, a machine learning model can be performed on such a representation for some down stream tasks. Here the term **target text** can be a word, a phrase, a sentence, a paragraph or a document. As introduced in Section 2.4, the representation of a target text can be obtained mainly through three methods: (1) Feature engineering methods based on manually defined features, such as bag-of-word, term frequency and inverse document frequency (TF-IDF), shifter, n-grams, POS tags, or count of emotion categories based on a given emotion lexicon [164, 153, 99]; (2) Distributional methods that treat a target text as single non-divisible unit and learn its representations directly from its context based on the distributional hypothesis [63, 94]; and (3) Compositional methods to infer the representation of a target text from its component words based on the principle of compositionality. Compared to feature engineering methods, both distributional and compositional methods are less labor intensive.

Phrases, as one kind of language units, play an important role in many NLP applications such as machine translation, web searching and sentiment analysis [107]. Generally speaking, phrases can be categorized as either **compositional** or **non-compositional**. For compositional phrases, such as *traffic light, swimming pool*, their semantics are composed from the semantics of their component words. We define component words as the **inter-**

nal context of a phrase. For non-compositional phrases, such as multiword expressions *couch potato* and *kick the bucket*, their semantics are generally not directly related to the semantics of their component words. According to [133], in a corpus with a collection of web pages, about 15% of word tokens belong to multiword expressions, 57% of sentences and 88% documents contain at least one multiword expression.

Chapter 4 presents word level emotion analysis and shows that the affective meanings of words can be effectively inferred from word embedding representation through a regression model.

In this chapter, we propose two methods for phrase level emotion analysis. The first method is based on word representations and composition models to perform emotion prediction. The second method directly learns the phrase embedding representation and then decodes the affective information from phrase embedding just like Chapter 4. The two methods will be introduced separately since they follow different strategies.

5.1 Composition Based Emotion Analysis

As presented in **Chapter 4**, multi-dimensional affective representations of words can be obtained by various methods. One advantage of dimensional affective representations over discrete affective representations is that dimensional vectors can be computed directly. The issue then is: if multi-dimensional emotion lexicons are used to infer emotions of larger target text, are the domain specific emotion lexicons have any advantage over other word representations?

This section attempts to answer this question by investigating the most effective ways to predict affective meaning of larger text units using dimensional word representations based on compositional models.

5.1.1 Composition Models for Emotion Analysis

The objective is to study the effectiveness of different word representations for emotion analysis of text units longer than words using compositional models. A general learning framework, as shown in **Figure 5.1**, is proposed for this investigation. In this framework, the **Word Representation**, which may contain either affective knowledge or general semantic knowledge, can be a one-dimensional sentiment lexicon, a multi-dimensional affective lexicon, or a general semantic based word embedding lexicon. The **Target Text**, as input data, can be any text that is composed of word sequences. The **Composition Model** can be any composition model introduced in **Section 2.4**, such as concatenation, addition, multiplication, or more complex LSTM models. The output, **Emotions**, should be emotion labels or affective values as a downstream task after their associated emotions are predicted by the **Emotion Prediction** module.

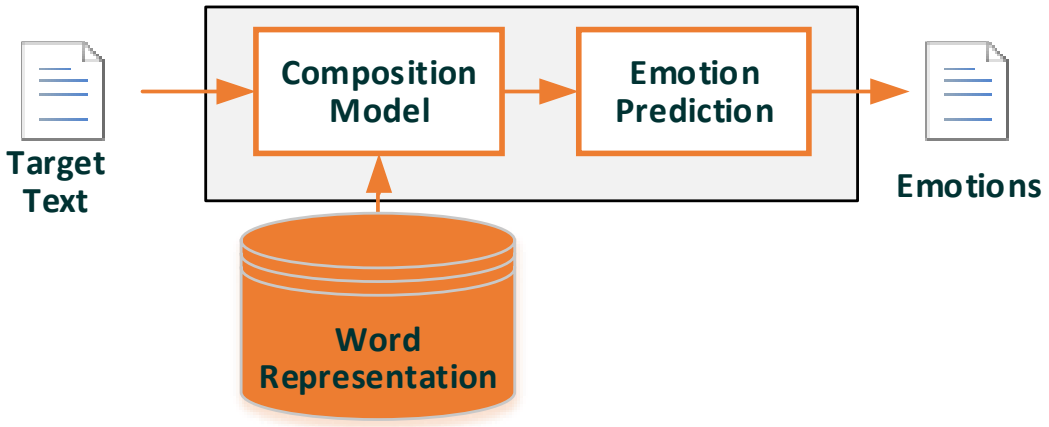


Figure 5.1: General composition framework for emotion analysis.

To focus more on the effectiveness of representations, experiments are conducted on bigram phrases. Five lexicons used in Chapter 4 are selected for this study, including three manually annotated lexicons introduced in Table 4.1 (VADER sentiment lexicon, the EPA lexicon, and the E-ANEW lexicon), one automatically obtained lexicon introduced in Table 4.16 (NRC Hashtag sentiment lexicon, denoted as **HSenti** here) , and one word

embedding lexicon introduced in Table 4.3.6 (Glove).

Note that manually annotated lexicons have much smaller size than automatically obtained ones. For fair comparisons, all the lexicons are extended by the method introduced in Chapter 4.2 based on the Glove word embedding so that all the vocabularies of different lexicons are the same size as Glove.

Firstly the representation of a phrase is constructed from the base representations of its component words using some composition functions. Here the term **base representations** refers to the different word representations used in this study. Then, the prediction for phrases is performed to see which base representation is more effective.

The following four composition models introduced in Section 2.11 are used.

1. The addition model defined in Formula 2.11.

2. The multiplication model is defined in Formula 2.13.

3. The concatenation model defined in Formula 2.14.

4. The LSTM model defined in Formula 2.19 is shown in Figure 5.2, which takes the phrase *avoid accident* as an example.

In Figure 5.2, the input is the vector representations of words *avoid* and *accident*. The output of LSTM is a vector $\vec{m}$. The emotion y is predicted based on $\vec{m}$.

5.1.2 Experiments and Analysis

A set of sentiment classification tasks are conducted. The gold answers for the sentiment classification of phrases are extracted from the Stanford Sentiment Treebank (SST) [139]. In SST, every sentence is parsed and each node in the parsed tree has a sentiment score ranging between [0, 1]. The sentences are movie review excerpts from the *rottentomatoes.com* website. Only two-word adjective-noun phrases, noun-noun phrases and verb-

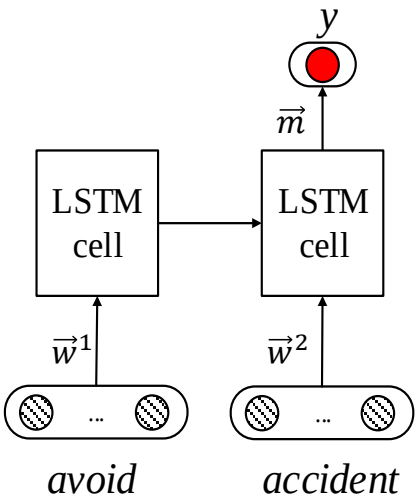


Figure 5.2: The LSTM compositional model for emotion analysis.

noun phrases are extracted, and the size of the collection in SST is 9,922. Note that only 6,736 phrases in this collection are used because they are present in all the five lexicons.

Based on this golden answer set on sentiments of phrases, four sentiment analysis tasks are constructed: (1) a regression task to predict the sentiment score of phrases (labeled as **SST-R**); (2) a binary classification task by converting sentiment scores to discrete labels, where positive label is no less than 0.6 and negative label is no more than 0.4 (labeled as **SST-2c**). The numbers of phrases that fall into the positive and negative classes is 2,669 and 1,539, respectively; (3) a ternary classification task similar to SST-2C except that there is an additional neutral label in the range of 0.4-0.6 (labeled as **SST-3c**). The numbers of phrases that fall into the negative, neural and positive classes are 1,539, 5,714, and 2,669, respectively; (4) a five-class classification task that segments the rating score by the following standard: [0, 0.2], (0.2, 0.4], (0.4, 0.6], (0.6, 0.8], (0.8, 1.0], which represents very negative, negative, neutral, positive, very positive respectively (denoted as **SST-5c**). The class distribution is 166, 1, 373, 5,714, 2,222, and 447, respectively.

Different evaluation metrics are used for the four different tasks. Root mean squared error (**rmse**), mean absolute error (**mae**) and Kendall rank correlation coefficient (τ) are

used for SST-R. **Accuracy** and **F-score** are used for SST-2c. **Weighted accuracy** and **weighted F-score** are used for SST-3c and SST-5c. Ridge regression and SVM with the linear kernel are used for regression and classification task, respectively.[1] For LSTM, the output layer is set differently for regression and classification tasks, respectively. In all the experiments, 5-fold cross validation is used. Results are based on the best parameters. The number of hidden dimensions in LSTM is set to 4.

Feature	**Comp**	**SST-R**			**SST-2c**		**SST-3c**		**SST-5c**	
		rmse	**mae**	τ	**acc**	**f**	**acc**	**f**	**acc**	**f**
HSenti	mul	0.110	0.110	0.060	0.636	0.777	0.573	0.418	0.573	0.418
HSenti	add	0.102	0.102	0.298	0.764	0.826	0.573	0.418	0.573	0.418
HSenti	conc	0.102	0.102	0.304	0.768	0.829	0.573	0.418	0.573	0.418
HSenti	lstm	0.100	0.100	0.307	0.769	0.825	0.609	0.554	0.583	0.470
VADER	mul	0.103	0.103	0.240	0.666	0.787	0.608	0.508	0.575	0.420
VADER	add	0.088	0.088	0.477	0.888	0.913	0.643	0.577	0.613	0.522
VADER	conc	0.086	0.086	0.482	0.889	0.914	0.654	0.590	0.621	0.532
VADER	lstm	0.085	0.085	0.487	0.895	0.918	0.668	0.657	0.618	0.552
EPA	mul	0.097	0.097	0.367	0.834	0.872	0.575	0.420	0.575	0.420
EPA	add	0.092	0.092	0.422	0.887	0.912	0.600	0.488	0.580	0.440
EPA	conc	0.092	0.092	0.427	0.886	0.912	0.602	0.494	0.588	0.463
EPA	lstm	0.092	0.092	0.436	0.893	0.916	0.637	0.611	0.600	0.507
E-ANEW	mul	0.099	0.099	0.313	0.769	0.830	0.601	0.497	0.575	0.420
E-ANEW	add	0.090	0.090	0.451	0.890	0.913	0.620	0.549	0.575	0.420
E-ANEW	conc	0.089	0.089	0.458	0.894	0.917	0.623	0.555	0.575	0.420
E-ANEW	lstm	0.088	0.088	0.471	0.902	0.924	0.653	0.643	0.613	0.564
Glove	mul	0.106	0.106	0.245	0.635	0.777	0.575	0.420	0.575	0.420
Glove	add	0.074	0.074	0.564	0.920	0.937	**0.757**	0.751	**0.700**	0.679
Glove	conc	0.074	0.074	0.563	0.924	0.940	0.755	0.749	0.699	0.680
Glove	lstm	**0.070**	**0.070**	**0.578**	**0.926**	**0.942**	0.754	**0.752**	0.698	**0.683**

Table 5.1: Performance of different word representations under different composition functions for phrase sentiment analysis.

Table 5.1 shows the result of the four tasks. **mul**, **add**, **conc** are for multiplication composition, addition composition and concatenation composition, respectively. Table 5.1 shows four major points. Firstly, multiplication performs the worst in all categories.

[1] Using the scikit-learn tool: scikit-learn.org/

On the other hand, LSTM, as a deep learning method, is the best performer. Addition and concatenation do have comparable performance and not too off from LSTM on SST-R and SST-2c. Secondly, for the two sentiment lexicons, VADER performs much better than HSenti. This may be because that VADER is manually annotated from crowdsourcing whereas HSenti is automatically obtained which contains more noise. Thirdly, for the two multi-dimensional affective lexicons, E-ANEW performs slightly better than EPA. It is surprising that the multi-dimensional lexicons perform even worse than the sentiment lexicon VADER even though the annotated size of E-ANEW (13,915) is much larger than VADER (7,502). This puts a question mark on the quality of annotation for multi-dimensional lexicon resources. However, then, a sentiment lexicon may be better suited for a sentiment analysis task. Fourthly, word embedding[2] performs much better than all the other representations. For instance, it achieves a relative improvement of 17.7% under τ for SST-R over the secondly ranked VADER representation. Different composition models except the multiplication for word embedding perform comparably. In principle, LSTM should have even more advantages if the text length is longer. In this study, the performance difference is not obvious because the tested phrases are only bigrams.

When manually constructed affective lexicons are extended automatically, more noise can be introduced. To eliminate that factor, an additional experiment using only a manually annotated lexicon is conducted. We use the largest original E-ANEW lexicon without extension to be compared with word embedding lexicon. In this case, the intersection of E-ANEW and word embedding lexicon has 3,908 words. A subset corpus of SST containing these words has 5,251 phrases. Five-fold cross validation is used on this dataset. The result is shown in **Table 5.2**. Again, word embedding lexicon achieves much better result than manually annotated VAD lexicon. If coverage issue is considered, word embedding has even more advantages.

[2] We also experiment on different word embedding dimensions including 50,100,200. All are better than the other lexicons.

Feature	Comp	SST-R			SST-2c		SST-3c		SST-5c	
		rmse	**mae**	τ	**acc**	**f**	**acc**	**f**	**acc**	**f**
E-ANEW	add	0.093	0.120	0.450	0.901	0.928	0.614	0.555	0.555	0.397
E-ANEW	conc	0.092	0.118	0.464	0.905	0.930	0.622	0.566	0.560	0.426
E-ANEW	lstm	0.093	0.118	0.474	0.903	0.930	0.626	0.616	0.585	0.503
Glove	add	0.075	0.098	0.574	0.926	0.946	0.762	0.757	0.700	0.683
Glove	conc	0.075	0.098	0.577	0.928	0.947	0.762	0.757	0.697	0.679
Glove	lstm	0.079	0.103	0.556	0.923	0.943	0.721	0.720	0.658	0.653

Table 5.2: Performance of manual E-ANWE and word embedding under different composition functions for phrase sentiment analysis.

In conclusion, this section shows that automatically obtained word embedding outperforms both manually and automatically extended dimensional lexicons including sentiment lexicons and multi-dimensional emotion lexicons on the task of phrase level sentiment analysis based on different composition models. Although affective lexicons based on emotion models that are backed by cognitive theories are built specifically for affective analysis, building them consumes too many resources and annotation quality may still be questioned due to added complexity. Through a downstream task of sentiment analysis of phrases, the conclusion can be safely drawn that the manually annotated special purpose emotion lexicons have no advantage over lexicons of word embedding obtained automatically no matter which compositional model is used.

5.2 Phrase Embedding Based Emotion Analysis

Section 5.1 shows that compositional methods can be used to predict larger text unit for sentiment classification based on the principle of compositionality. However, for some phrases, their meanings cannot be inferred from their component words, such as *break up*, or idioms such as *kick the bucket* [64]. Generally speaking, phrases can be categorized as either **compositional** such as *traffic light, fresh air*, whose semantics are composed from the semantics of its component words, or **non-compositional** such as idiomaticity *couch*

potato and *kick the bucket*, whose semantics are not directly composed from its component words. Furthermore, Chapter 4 shows that affective meanings of words and phrases can be effectively inferred from their embeddings through a regression model. The technical question is: can we directly learn embedding representations of phrases and then infer the affective meanings of phrases from phrase embedding?

Generally speaking, there are mainly two approaches to learn the embedding of phrases. The first approach, referred to as the **distributional approach**, is based on the distributional hypothesis. To apply this principle for phrases, a phrase is simply treated as a single term and its representation is inferred from its external context in the same way as learning distributed word representation [94, 181]. Since this approach treats a phrase as a non-divisible unit, its component words are completely ignored even though this information may be useful, especially for compositional phrases. For example, the phrase *close interaction* is semantically similar to *contact*, which can be reflected through the word embedding similarities between the component word *interaction* and *contact*. However, by treating *close interaction* as one non-divisible unit, its representation has to be learned independently which is more likely to suffer from data sparseness problem. In the case of infrequently used phrases which would have insufficient context, this approach can fail to learn their representations. Compared to single words, the sparseness problem of phrases is indeed more severe given the same corpus.

The second approach, referred to as the **compositional approach**, is based on the principle of compositionality. Based on this principle, this approach uses certain composition function to obtain the representation of a phrase from the representations of its component words [97, 185], as is discussed in Section 5.1. The representation of the component words is obtained using distributional models. This approach only uses information of component words and the information of external context is indirectly considered using the context of the component words. One key problem with the compositional approach is that it can fail if a phrase is non-compositional because the semantics of non-compositional

phrase cannot and should not be derived from its component words. For example, the phrase *monkey business* cannot be composed from *monkey* and *business*. In such a situation, the information of the component words can lead to an erroneous result. The meaning of non-compositional phrases can be lost using compositional models.

In fact, both the external context and the component words provide helpful information to the representation of a phrase. Furthermore, the usefulness of component words depends on the compositionality of the phrase. **Compositionality** refers to the extent of the semantic of a phrase can be inferred from the semantic of its component words [128]. For example, the compositionality of phrase *monkey business* should be low and the compositionality of phrase *buy fruits* should be high. If there is a way to measure the compositionality of a phrase, the compositionality can then be used to measure the usefulness of the component words of a phrase. Based on the above analysis, a hybrid model is proposed in this work to learn the representation of phrases from both the external context and component words. Instead of simply combining the two kinds of information, the compositionality measures from lexical semantics are used to serve as a constraint. The basic idea is to learn the representation of a phrase based on a linear combination of external context with a weighted composition of the component words where the weight is based on automatically predicted compositionality. Compared to previous works, the proposed model has the advantages of both previous methods while overcomes their drawbacks. After obtaining the embedding of phrases, phrase related semantic information such as emotion and sentiment can be inferred from the embeddings.

5.2.1 Related Work

Compositionality of phrases includes two main tasks: compositionality detection and compositionality prediction. Compositionality detection aims to identify if a given phrase is compositional or not, which is considered a binary classification task. Yazdani et al. [180] propose a semantic composition based method for compositionality detection. Noun

compounds that cannot be well modeled by a compositional model are considered non-compositional. Compositionality prediction aims to predict the compositionality value of phrases in the range of [0, 1]. Salehi et al. [128] propose to compute the compositionality as the cosine similarity between the representation learned from external context and the representation composed from the phrase's component words.

As introduced in Section 2.3, the distributed representation of words can be learned by counting-based methods and prediction-based methods [10]. Both methods are based on the distributional hypothesis [49]. Similarly, to learn the representation of phrases, one approach treats a phrase as a single unit and learns phrase representations from its external context using the same word representation learning model [94, 181]. In this approach, a phrase is treated as a non-divisible unit. The second approaches treat phrases as divisible units and infer the representation from the representations of the component words, which is called compositional approach. In addition to the basic compositional models introduced in Section 2.4, Baroni et al. [11] propose to represent a noun as a vector and an adjective as a matrix and use matrix-vector multiplication to obtain the representation of adjective-noun phrases. Yu et al. [185] propose to obtain phrase representations by the weighted sum of word vectors and the weights are based on a list of lexical feature templates of the phrase types. Zhao et al. [189] propose a tensor based compositional model to learn phrase representations by vector-tensor-vector multiplication. However, all the above composition based methods assume that phrases are compositional. The study by Sun et al. [147] on phrase representation learning actually makes use of both external context and component words by constraining the vector of a phrase to be close to the vectors of both its two component words. However, the semantics of some phrases are not necessarily similar to both of its component words. Some non-compositional phrases have no direct relation to its component words. The work from [50] considers both the external context and component words with compositionality constraint. However, the learning process of that work is task dependent and that work only handles verb-noun phrases.

5.2.2 The Hybrid Model

Most of the works on learning phrase representation either only consider the external contexts which ignore the information of component words or only consider the component words without taking into account the information of external context and the compositionality of the phrases. In this section, a hybrid phrase representation learning model is proposed to include both the external context as well as the internal component words, similar to that of [50]. The main difference with [50] is that the proposed method uses a general compositionality constraint when merging the two components together. This method is not limited to verb-noun phrases. The learning model consists of two components. The first component is based on the distributional model of SGNS [94] to learn both word and phrase representations using external context. The second component is the compositional model to learn phrase representations from component words. This proposed hybrid model is referred to as the **D&C** (**D**istributional and **C**ompositional). However, in D&C, the two components are not simply added linearly. Instead, the compositional component is subjected to compositionality estimation with which the constrained compositional model is added to the distributional component. Similar to previous works, this work also focuses on phrases that consist of two component words. The following paragraph introduces some notations for this section.

Given a corpus S with a set of words $w \in V_W$ and their context $c \in V_C$ where V_W and V_C are the word and context vocabularies. Note that the vocabularies of V_w and V_C may be identical. The distinction is more for conceptual convenience. The context of word w_i is defined as the words surrounding w_i in a window of size L, namely $w_{i-L}, \cdots, w_{i-1}$,(w_i), $w_{i+1}, \cdots, w_{i+L}$. Let $\#(w)$ denote the frequency of word w, and $\#(c)$ denote the occurrence frequency of context c. Let $\#(w, c)$ denote the frequency of a word-context pair (w, c). For phrases,let V_M denote the set of given phrases where each phrase $m \in V_M$ consists of two words. t_m is used to denote the compositionality of m and the larger t_m is,

the more compositional is the phrase. Let D denote the set of (w, c) and (m, c) pairs.

The objective is to learn a vector representation $\vec{w} \in \mathbb{R}^d$ for each $w \in V_W$, a vector representation $\vec{c} \in \mathbb{R}^d$ for each context $c \in V_C$, and a vector representation $\vec{m} \in \mathbb{R}^d$ for each $m \in V_M$. d is the vector dimension.

The Distributional Component

The distributional component of a phrase representation is defined by the the SGNS model, introduced in Section 4.2.1. When applying this model to representation learning of phrases, $m_i \in V_M$ is treated as a single term and representation learning is the same as word representation learning. However, data sparseness can be an issue for phrases. Note that for compositional phrases, the SGNS component only takes information from external context of phrases. Context of component words is not directly taken into consideration.

The Compositional Component

As introduced in Section 2.4, in a compositional model, the representation of a phrase is inferred from that of its component words. Given a phrase m with two component words w_m^1 and w_m^2 and their respective vector representations $\vec{w}_m^1$ and $\vec{w}_m^2$, the representation of m, denoted by $\vec{m}$, can be computed by any simple compositional model defined from Formula 2.11 for addition to the Formula 2.13 for multiplication. Compared to the distributional component using SGNS, the compositional component can make use of component words information. However, this model can produce erroneous representation for non-compositional phrases.

The Hybrid D&C Model

The hybrid D&C model first makes use of an estimation on the compositionality of phrases. This estimation then serves as a constraint on the combined use of the distributional component of SGNS and the compositional component.

Since learning include both words and phrases, the candidate term set $V_T = V_W \cup V_M$ is first constructed and then the corresponding context set V_C is built based on the window size L. For a word w, its representation can be learned according to **Formula 4.5**. For a phrase m, the proposed D&C model can be modeled as:

$$J_S = \sum_{m \in V_M} \sum_{c \in V_C} \#(m,c) \Big(log\sigma(\vec{m} \cdot \vec{c}) \quad + k \cdot \mathbb{E}_{c_N \sim P_D}[log\sigma(-\vec{m} \cdot \vec{c_N})] \\ + \lambda h(t_m, \vec{m}, \vec{w}_m^1, \vec{w}_m^2) \Big). \tag{5.1}$$

In **Formula 5.1**, the first two parts forms the SGNS model, namely the distributional component. The third part is the compositional component. The hyper-parameter λ is a weight to balance the overall contributions of the two components. The compositional component h is defined as

$$h(t_m, \vec{m}, \vec{w}_m^1, \vec{w}_m^2) = t_m log\sigma \left(\vec{m} \cdot f(\vec{w}_m^1, \vec{w}_m^2) \right), \tag{5.2}$$

where $f(\vec{w}_m^1, \vec{w}_m^2)$ can be any compositional model defined by **Formula 2.9**. $\sigma\left(\vec{m} \cdot f(\vec{w}_m^1, \vec{w}_m^2)\right)$ defines the correlation between the learned phrase representation $\vec{m}$ and the composed phrase representation. The more they are correlated, the larger contribution the compositional component is to J_S. t_m is the compositionality of m, which is dependent on the phrase in questions. **Formula 5.2** has the following properties:

1. If the compositionality t_m is low (m is more non-compositional), the weight of the correlation between the phrase representation $\vec{m}$ and the composed representation $f(\vec{w}_m^1, \vec{w}_m^2)$ from its component words should be low. It means $\vec{m}$ should be based mainly on SGNS, namely its external context.

2. If the compositionality t_m is high (m being more compositional), the weight of the correlation between $\vec{m}$ and $f(\vec{w}_m^1, \vec{w}_m^2)$ should be high and the objective function

will force $\vec{m}$ to be similar to the composed $f(\vec{w}_m^1, \vec{w}_m^2)$. It means $\vec{m}$ should consider both the external context and component words.

By setting λ to zero, the model degrades to the SGNS model. By setting t_m to a constant, the model changes to a fix-weighted model.

Compositionality prediction

One of the most important elements of D&C is the compositionality value t. The compositionality prediction model aims to predict the compositionality of a phrase. Phrase compositionality has the property of continuum [124]. For example, the compositionality of phrase *bus driver* is 1.0, which means this phrase is compositional and the meaning of it can be composed from the component words *bus* and *driver*. The compositionality of phrase *coach potato* is 0, which means this phrase is non-compositional and the meaning of it cannot be inferred from the component words *coach* and *potato*. The compositionality of the phrase *silver screen* is 0.6, which indicates that its semantics cannot be totally obtained from the component words because the first word *silver* loses its original meaning in the phrase while the second word *screen* can reflect the phrase' meaning. In this section, we introduce two models for predicting individual compositionality of phrases.

The first model is from [128], which computes the compositionality of a phrase based on the consine similarity between the distributional embedding and the compositional embedding of the phrase defined as:

$$t_m = cosine\left(\vec{m}, \vec{w}_m^1 + \vec{w}_m^2\right), \tag{5.3}$$

where $\vec{m}$, $\vec{w}_m^1$ and $\vec{w}_m^2$ are obtained by SGNS in advance. Formula 5.3 means that the more similar between $\vec{m}$ and $\vec{w}_m^1 + \vec{w}_m^2$, the more compositional the phrase is. We label this compositionality prediction model as **C1**.

The second model is inspired by the work from [43] which is based on the geometry of word embedding. They find that the semantic space of larger text units (such as phrases and

sentences) is spanned by the subspace of the consisting word vectors and the subspace can be obtained through dimension reduction such as Principle Component Analysis (PCA). Inspired by this, we propose to compute phrase compositionality by computing the cosine similarity between the distributional embedding and the projected vector on the subspace spanned by the component word embeddings. The process is shown in Figure 5.3. Given a phrase m consisting of two words w_m^1 and w_m^2, $\vec{m}$ is the distributional phrase embedding and $\vec{v}_m^1$ and $\vec{v}_m^2$ are the distributional component word embedding, obtained by distributional methods. $\vec{m}_p$ is the projected vector of $\vec{m}$ on the space spanned by $\vec{v}_m^1$ and $\vec{v}_m^2$. Let $A = [\vec{v}_m^{1T}, \vec{v}_m^{2T}]$. $\vec{m}_p$ is computed as:

$$\vec{m}_p = A(A^T A)^{-1} A^T \vec{m}. \tag{5.4}$$

The compositionality is computed as:

$$t_m = cosine(\vec{m}, \vec{m}_p). \tag{5.5}$$

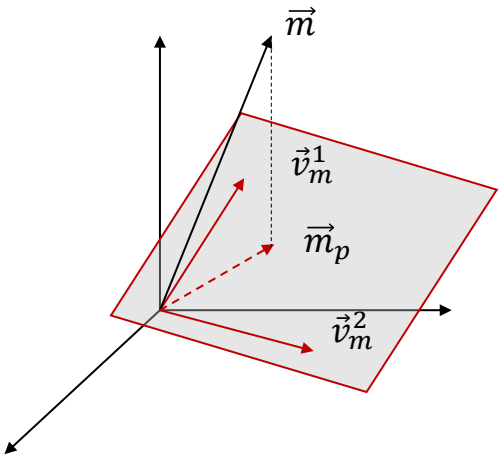


Figure 5.3: The C2 model for compositionality prediction.

This compositionality prediction model is denoted as **C2**. Compared to C1, C2 assumes that if a phrase is compositional, its phrase representation is the subspace spanned by its component words. The more the distributional embedding is close to the subspace,

the more compositional of the phrase. If the distributional embedding is perpendicular to the subspace, the phrase is non-compositional.

Theoretically speaking, any phrase compositionality model can be used in our proposed framework. Note that the compositionality values of phrases are computed based on the distributional embedding before training the model.

Our model can be trained through stochastic gradient descent (SGD) by maximizing Formula 5.1 suggested by [94]. The gradient can be directly calculated for each training sample. Both the word embeddings and phrase embeddings are randomly initialized as what is used by Mikolov et al [94].

5.2.3 Experiments and Analysis

The list of phrases used in the evaluation are from 5 sources: (1) the set of 2,180 phrases in the Noun-Modifier Composition dataset [159], (2) the DISCo set of 349 phrases for the 2011 shared task in Distributional Semantics and Compositionality [14], (3) the set of 8,105 phrases from the SemEval 2013 Task 5A [65], (4) the set of 1,042 phrases from [34], and (5) the set of 56,850 phrases from [181]. The consolidated phrase list contained has a total of 60,315 after removal of duplication.

The training corpus used is the Wikipedia August 2016 dump.[3] In pre-processing, pure digits and punctuations are removed, and all English words are converted to lowercase. The final corpus consists of about 3.2 billion words. During training, only words that occur more than 100 times are kept, resulting in a vocabulary of 204,981 words.

Evaluation Tasks

The representation of phrases is evaluated on four evaluation tasks. The first task is called the **SemEval 2013 Task 5**. The dataset for this task, denoted as **SemEval**, is prepared to judge whether a given bigram-unigram pair is semantically related or not [65]. For

[3] https://dumps.wikimedia.org/enwiki/latest/

example, the bigram *newborn infant* is semantically related to the unigram *neonate*. So, the gold answer for this pair is *(newborn infant, neonate, 1)*, where the label *1* indicates their relatedness. On the other hand, the bigram *stable condition* is not related to the unigram *interview*, so in the gold answer the entry is *(stable condition, interview, 0)*. The officially released data contains 7,814 test samples and 11,722 training samples.[4] Since some of the bigrams/unigrams are not contained in the Wikipedia training corpus, only 15,973 samples contained in Wikipedia are used for evaluation. Since SemEval 2013 Task 5 is a binary classification problem, cosine similarity between learned bigram embedding and the unigram embedding is used as the feature. SVM is used to learn the threshold for the classification based on 5-fold cross-validation. Accuracy, precision, recall and F-score are used in the evaluation metrics.

The second task is called **Phrase Similarity**, denoted as **PS**. This task provides a phrase pair similarity dataset with 324 samples[5] constructed using manually rated scores from 1 to 7 with 7 being the most similar [97]. For example, the phrase pair *(hot weather, cold air)* has a similarity score 2.22. The dataset contains three types of phrases: adjective-nouns, noun-nouns, and verb-objects with 108 samples for each type. All 324 samples are used in evaluation. Cosine similarity is used to compare different phrase vectors and Spearman's ρ correlation coefficient is used to evaluate the performance.

The third task is called **Turney-5**, denoted as **T-5**. The dataset in this task is a 7-choice Noun-Modifier Question dataset built from WordNet [159] with 2,180 question groups. For example, in the sample *(small letter, lowercase, small, letter, little, missive, ploughman, debt)*, the first bigram *small leter* is the question and the latter 7 unigrams are the candidate answers. The task is to select the most similar unigram as the answer, which should be *lowercase* in this sample. To remove the bias towards component words by following Yu's suggestion [185], the two component words are removed to construct a

[4] https://www.cs.york.ac.uk/semeval-2013/task5.html.

[5] http://homepages.inf.ed.ac.uk/mlap/index.php?page=resources

5 choice single word question to form the evaluation dataset. Again, by removing samples that are not contained in the Wikipedia training corpus, the final evaluation data contains 669 questions. Cosine similarity is used to measure the semantic closeness of a bigram phrase and the unigrams. The one with the highest similarity score is chosen as the answer. Accuracy is used as the evaluation metric.

The fourth task predicts the sentiment of phrases, the same task conducted as in Section 5.1. The used corpus are the phrases extracted from the Stanford Sentiment Treebank (SST) with each phrase annotated with a sentiment score from 0 to 1. The overlapping set of the phrases in SST and the phrases in phrase embedding is 772. The target is to predict the sentiment score of the 772 phrases, which is a regression problem. This corpus is denoted as **SST**. The evaluation metrics include Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and the Kendall rank correlation coefficient (τ). For RMSE and MAE, the smaller of the value is, the better of the performance is, vice versa for τ. Similar to Chapter 4, the Ridge Regression model is used to predict the sentiment score of the phrases from the phrase embedding.

Baselines and Experiment Settings

The D&C model is compared to the following baselines:

1. **SGNS**: the original vector representation model to take a phrase as a non-divisible unit [94, 181];

2. **SEING**: a modified SGNS model by treating component words as *morphemes* of the phrase with a constraint that the phrase vector should be similar to both the vectors of its the component words regardless of the compositionality of the phrase [147];

3. **Comp-Add**: a simple addition compositional model to use the sum of the vectors of the component words as the phrase's vector.

4. **Comp-Mul**: a simple multiplication compositional model to use the multiplication of the two component vectors to obtain the phrase's vector.

5. **Comp-W1**: a compositional model to use the vector of the first component word directly as the vector of a phrase.

6. **Comp-W2**: a compositional model to use the vector of the second component word directly as the vector of a phrase.

The proposed D&C model has three settings for compositionality t_m. The first one directly sets t_m as a constant, $t_m = 1$, denoted as **D&C-*C***. This means the compositionality of all phrases is set fixed as an identical and fixed number. The second one uses automatically computed t_m by model C1, denoted as **D&C-*C1***. The third one uses automatically obtained t_m by model C2, denoted as **D&C-*C2***. Both D&C-C1 and D&C-C2 estimate compositionality for each phrase individually.

The size of the context window for all the models is set to 5, negative samples size is 5, and the vector dimension is 300. λ is empirically set to 8. For the compositional model, we empirically evaluate several kinds of combinations such as the addition model with α and β as 1, or the multiplication model. Experiments show that the addition compositional model achieves the best, so only the results using the addition model are reported here. To obtain the compositionality t_m, firstly the representation of phrases is learned using SGNS and its compositionality is computed based on C1 and C2, respectively.

The evaluation result on the four datasets under different evaluation metrics is shown in Table 5.3. In this table, the first two models are distributional methods, the middle four models are compositional methods and the last three models are three variants of the proposed model. The percentage after each dataset name is the proportion of non-compositional phrases in that dataset. The percentage is obtained by randomly sample 30 phrases in each data set and then manually verify their compositionality. We can see

Model	**SemEval** (2.5%)				**PS** (2.5%)	**T-5** (10%)	**SST** (30%)		
	Acc	**Pre**	**Rec**	**F**	ρ	**Acc**	**rmse**	**mae**	τ
SGNS	.629	.728	.412	.526	.155	.535	.094	.063	.218
SEING	.586	.562	.773	.651	.056	.576	.089	.061	.269
Comp-Add	**.795**	**.826**	**.748**	**.785**	**.622**	.603	.090	.066	.283
Comp-Mul	.506	.506	.483	.494	410	.227	.098	.063	.219
Comp-W1	.737	.771	.672	.718	.450	.499	.092	.065	.211
Comp-W2	.759	.796	.697	.743	.500	.463	.100	.071	.113
D&C-C	**.779**	.808	**.731**	**.767**	.595	**.683**	.089	**.061**	**.301**
D&C-*C1*	.764	.794	.711	.750	.580	**.681**	**.087**	**.060**	**.310**
D&C-*C2*	.776	**.**841	.681	.753	**.623**	.677	**.088**	.061	.293

Table 5.3: Performance of different phrase representation learning models. The top two performers are in bold and the best performer is also underlined.

that different datasets do have different proportions of non-compositional phrases and this should have effects to the performance of different methods.

General Analysis

Comparison between distributional methods and compositional methods shows that compositional methods achieve much better result than distributional methods. For example, on SemEval, Comp-Add achieves a relative improvement of 49.2% under F-score compared to SGNS. In other words, the semantics of phrase expressions are not fully recognized by using only external context. Treating phrases as a non-divisible units obviously loses some semantic information carried by the component words. This also indicates that in a real application, compositional models are a better choice compared to a distributional approach for phrase embedding learning. Comparing between distributional models, SEING performs better than SGNS on SemEval, T-5 and SST. But, SEING performs worse than SGNS on PS. Further analysis of SEING on PS indicates that the cosine similarities of many phrase pairs in PS are negative. Among the four baseline compositional methods, Comp-Add performs much better than other compositional methods. Comp-Mul performs the worst. This means that element-wise multiplication can introduce more noise than

information. Comp-W1 and Comp-W2 have similar performance with Comp-W2 performing slightly better on SemEval and PS and Comp-W1 performing better on T-5 and SST. Among all the models, Comp-Add performs the best on the SemEval dataset while our proposed model D&C performs the best on PS, T-5 and SST. Specifically, on T-5, the best performer D&C achieves a relative improvement of 13.3% over Comp-Add. This indicates the effectiveness of our proposed model. Among the three variants of D&C, no one is overall best. D&C-C performs the best on SemEval and T-5, while D&C-C2 performs the best on PS. D&C-C1 performs the best on SST under rmse, mae and τ. Overall, our proposed model achieves the most robust result since D&C is always the top two performer on all datasets and in fact top performer in three out of four datasets.

Further analysis indicates that the performances of different models are dataset dependent, especially dependent on the proportion of non-compositional phrases. As shown in Table 5.3, the proportions of non-compositional phrases are 2.5%, 2.5%, 10%, and 30% in SemEval, PS, T-5 and SST, respectively. Because compositional models are more suitable for compositional phrases, Comp-Add performs much better than SGNS on SemEval. However, the gap decreases on T-5 between SGNS and Comp-Add as the proportion of non-compositional phrases increases. Performance of Comp-Add indicates that the combined use of the vectors of two component words is more comprehensive than using external contexts for compositional phrases. On T-5 and SST datasets, the proportions of the non-compositional phrases are larger than in the other two sets. So, there are more phrases which would not work using compositional methods. That is why the performance of SGNS increases and D&C outperforms Comp-Add.

Compositionality Analysis

To further explore the effects of compositionality on different methods, the proportion of non-compositional phrases are further analyzed based on the SemEval semantic relation task. 20 non-compositional phrases are manually selected from Farahmand's list which

has 1,042 phrases manually annotated with compositionality values [34]. Each phrase is annotated by four annotators with 1 indicating non-compositional and 0 as compositional. Based on the 20 phrases, 20 positive (semantically related) bigram-unigram pairs and 20 negative (not semantically related) bigram-unigram pairs are constructed to form a balanced non-compositional sample set for the SemEval task, denoted as **N-Sem**. 60 samples from the original SemEval dataset are also taken to form a compositional sample set, denoted as **C-Sem**. In the evaluation, the non-compositional phrases from N-Sem are added to C-Sem to increase the proportion of non-compositional phrases until all the non-compositional phrases are used up (total of 100 samples). Then the compositional portion is reduced so that the non-compositional proportion reaches about 70% of the total set (57 samples). The two distributional models, SGNS and SEING, are selected for evaluation. Since Comp-Add performs much better than the other three compositional models, only Comp-Add is included for comparison. For comparison, I introduce another variant of D&C, **D&C-M**, which uses manually annotated compositionality as t_m, which is obtained as follows. I first obtain the sum the four annotation values as a and convert a by $t_m = (4 - a)/4$ to obtain t_m as the gold compositionality value. t_m is in the range of [0,1] and is consistent with our definition of compositionality (namely 1 indicates compositional, 0 indicates non-compositional). F-score is used as the evaluation metric. Because of the limited data size, each model is run 10 times and the average is used.

The result is shown in **Figure 5.4**. This figure shows that when the proportion of non-compositional phrases is small, Comp-Add performs much better than SGNS, consistent with the result in **Table 5.3**. As the non-compositional portion increases, the performance of Comp-Add degrades gradually whereas in contrast, the performance of SGNS increases gradually. This indicates that external context is indeed useful for non-compositional phrases and the compositional model is ill-suited for non-compositional phrases. The performance of SEING indicates that the constraint to force a phrase's vector to be similar to both of its components can actually bring adverse effect for non-compositional

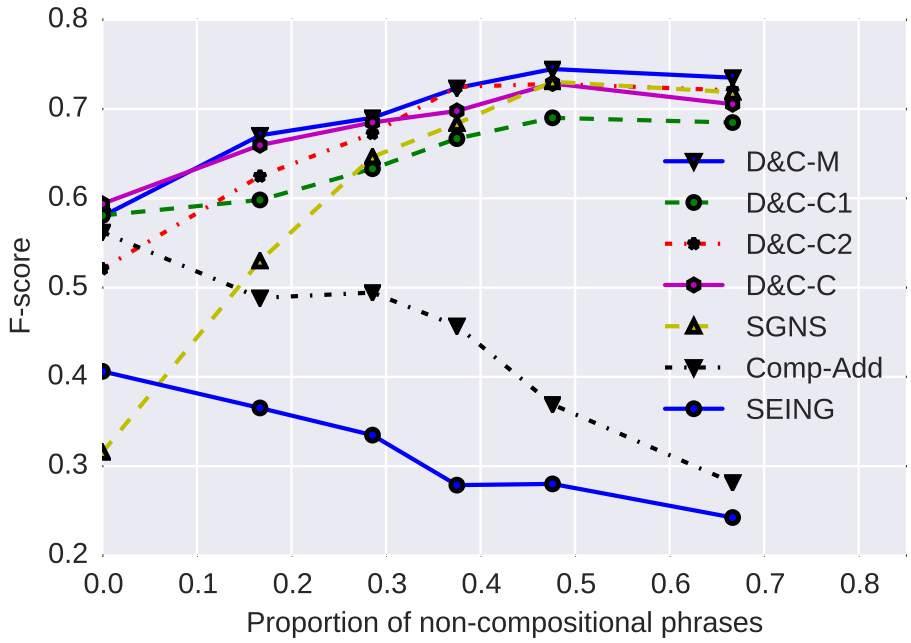


Figure 5.4: Performance of increasing the proportion of non-compositional phrases.

phrases. Over the whole spectrum, D&C gives a much more stable performance and is the overall top performer in all the automatic methods. D&C-M, which uses manually annotated compositionality, gives the best performance. The better performance of D&C-M over D&C-*C1* and S&C-*C2* indicates that there is still room for improvement on compositionality estimation. To validate this, a selected group of phrases are evaluated from Farahmand's list [34]. The overlapping of the phrase list with our phrase list is 408. The 408 phrases is used to evaluate the performance of the two compositionality prediction models. The estimated compositionality values by model D&C-C1 and D&C-C2 are compared with the gold compositionality by calculating Spearman's ρ correlation between the gold compositionality and the estimated compositionality. The result shows that ρ only achieves 0.227 and 0.200 for compositionality prediction model C1 and C2 respectively, which means the current method for compositionality estimation still has much room for improvement.

Hyper-parameter Analysis

To investigate the effects of the hyper-parameter λ, Figure 5.5 shows the effects of the weight λ on different tasks, which indicates that D&C-C achieves the best performance when λ equals 8.

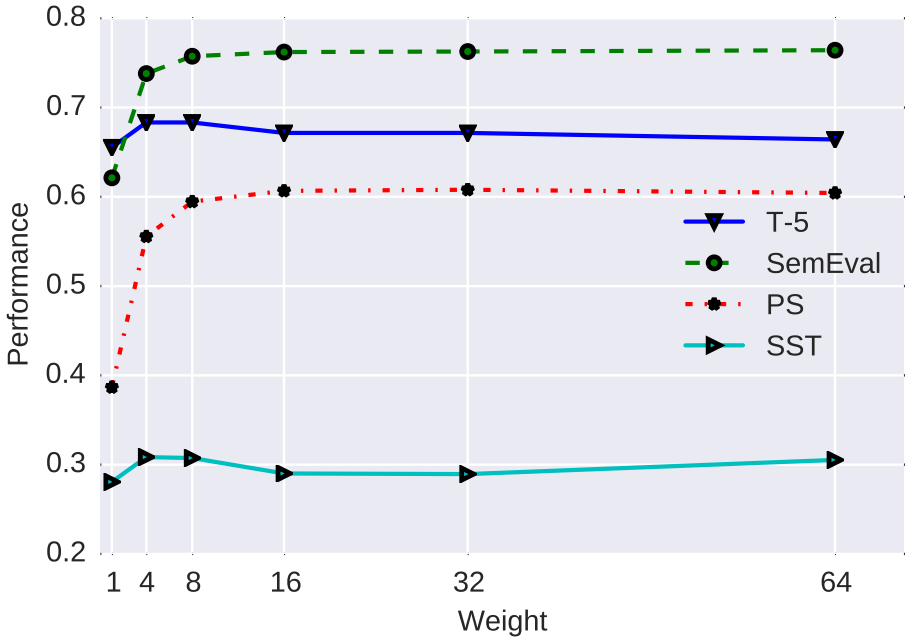


Figure 5.5: Performance of D&C-C with different λ values.

Case Study

To examine the performance of each model more closely, we select four phrases, (*swimming pool*, *game plan*, *melting post*, *rat run*) to extract the top 5 most similar words by different models. The phrases are selected based on their occurrence frequencies and the compositionality values. According to [124], their compositionality values are in the two ends of spectrum from 0 to 5 with 0 indicating the most non-compositional and 5 indicating the most compositional. The statistics of the four phrases are shown in Table 5.4. **Frequency** is their occurrence frequency in the Wikipedia corpus and **Compositionality**

is the annotated value by [124]. As shown in Table 5.4, the first phrase, *swimming pool* is highly compositional with high frequency of occurrence. The second phrase, *game plan*, is highly compositional but low in frequency. The third phrase, *melting pot*, is low in compositionality yet high in frequency. The last phrase, *rat run*, is low both in compositionality and in frequency. Top 5 most similar words/phrases are listed based on the cosine similarity between the phrase embedding and the word/phrase embedding. The result of selected words/phrases based on different models is shown in Table 5.5. Overall, Comp-Mul gives the most unreasonable results. Comp-W1 and Comp-W2 give results similar to the first component word and the second component word, respectively. So they are put aside in the following discussion.

	swimming pool	game plan	melting pot	rat run
Frequency	17794	116	6119	4
Compositionality	4.87	3.83	0.54	0.79

Table 5.4: Statistics of the selected example phrases.

For the phrase *swimming pool*, which has the high compositionality and high frequency, all the models give reasonable results that are semantically similar to *swimming pool*. For the phrase *game plan*, which is high in compositionality and low in frequency, the results from SGNS are not reasonable. For example, all the result phrases *fool up, run book, make book luck out*, and *times sign* are not closely related to *game plan*. This is obviously consistent with our claim that SGNS cannot perform well when the occurrence frequency is low. SEING gives reasonable results because it constrains a phrase to be semantically related to its component words. Comp-Add also gives semantically related words/phrases although most of the them are related to the word *game*. The three variants of the proposed D&C model all give similar results including the most reasonable phrase *game plans*. For the phrase *melting pot*, which is low in compositionality and high in frequency, SGNS gives reasonable results, which are all related to politics. On the contrary,

both SEING and Comp-Add give unreasonable results as they are related to either the component word *pot* or *melting* but not related to *melting pot*. Again, the three variants of our proposed D&C model give reasonable results as they are all related to *melting pot*. For the phrase *rat run*, which is low in compositionality and low in frequency, results given by all the models are not quite suitable. This is because all the distributional models fail under low frequency. On the other hand, all the compositional models fail because the phrase is non-compositional. However, our proposed models still gives one semantically related phrase *rat running*.

In conclusion, this case study validates that distributional models will fail when the occurrence frequency of a phrase is low and compositional models will fail when a phrase is non-compositional. Our proposed model gives the most robust answers. However, none of the models perform well when a phrase is non-compositional with low occurrence frequency.

Obviously, a distributional model performs better than a compositional model when the proportion of non-compositional phrases is large and a compositional model performs better when the proportion of non-compositional phrases is small. However, in practice, we do not have prior knowledge on the proportion of non-compositional phrases. This is why our proposed method has its advantage over both models individually as our method learns compositionality for individual phrases. Consequently, D&C is less sensitive to datasets, especially the proportion of non-compositional phrases. This is the reason that D&C has an overall better performance and more robust no matter what proportion of non-compositional phrases an application has. In addition, the fact that D&C-M gives better performance than D&C-C1 and D&C-C2 indicates that manually annotated compositionality is more reliable than predicted compositionality. This highlights the need for a more accurate compositionality estimation method.

Models	**swimming pool**	**game plan**	**melting pot**	**rat run**

SGNS	*swimming pools, squash courts, tennis courts, climbing wall, basketball courts*	*fool up, run book, make book, luck out, times sign*	*diasporic, middle eastern, mestizaje, caribbeans, ethnicities*	*holds true, faster computers, improve understanding, fuzzy set, molecular entity*
SEING	*swimming pools, pool hall, pool halls, tennis courts, wading pool*	*strategy game, arcade game, saved game, strategy games, game board*	*cooking pot, pot luck, pot roast, coffee pot, pot shots*	*hog line, hoosier state, blade roast, w byrd, running dog*
Comp-Add	*swimming, swimming pool, squash courts, pools, swimming pools*	*game, the game, plans, a game, strategy game*	*pot, melt, cooking pot, saucepan, boiling*	*rat, brown rat, roof rat, black rat, giant kangaroo*
Comp-Mul	*weberian, individuation, apparatuses, cope, internalization*	*negatives, barb, stag, andersons, smallville*	*pot, cooking pot, talgai, pocket knife, pinfold*	*controversially, sion, furthered, controversy, tahiti*
Comp-W1	*swimming pool, aquatics, swim, synchronized swimming, squash courts*	*the game, games, card game, video game, wiiware*	*melt, melts, melted, melting point, eutectic*	*rats, rodent, rattus, mole rat, muridae*
Comp-W2	*pools, swimming pool, squash courts, wading pool, swimming pools*	*plans, planning, master plan, planned, proposal*	*pots, cooking pot, saucepan, pourri, ladle*	*running, runs, ran, run in, run on*
D&C-C	*swimming pools, tennis courts, squash courts, basketball courts, fitness center*	*game plans, a game, saved game, end game, waiting game*	*diasporic, mestizaje, caribbeans, middle eastern, folk culture*	*rat running, rat through, rat on, rat trap, young rat*
D&C-C1	*swimming pools, squash courts, tennis courts, basketball courts, fitness center*	*game plans, end game, waiting game, saved game, game board*	*diasporic, mestizaje, caribbeans, diasporas, folk culture*	*rat running, rat through, rat on, rat race, rat trap*

D&C-C2	*swimming pools, tennis courts, squash courts, basketball courts, indoor pool*	*game plans, the game, a game, strategy game, board game*	*diasporic, mestizaje, caribbeans, ethnicities, diasporas*	*rat running, rat through, rat on, rat, rat trap*

Table 5.5: The top 5 similar words of four kinds of phrases.

5.3 Chapter Summary

In this chapter, two pieces of work are conducted for phrase level emotion analysis. The first work investigate the performance of combinations of different word representations and composition models. The experiment result shows that multi-dimensional affective lexicons do not have advantages over automatic word embedding. The second work is based on phrase embedding learning to infer the emotional information. A hybrid model, D&C, is proposed to learn the embedding representation of phrases from their external context and component words with the compositionality constraint. This model can make use of both the external context and component words of phrases. Instead of a simple combination of the two kinds of information, compositionality measures from lexical semantics are used to serve as a constraint. Evaluations on four phrase semantic analysis tasks show that the proposed model performs better than both compositional methods and distributional method regardless of the proportion of non-compositional phrases in the dataset. As compositionality measure is introduced, the proposed hybrid model is the most robust on both compositional and non-compositional phrases, which also indicates that incorporating more semantic information brings benefits for representation learning. The D&C model performs much better than the baselines on the phrase sentiment score prediction task. Even though the model gives a theoretically sound solution, the compositionality estimation method still has room for improvement. In the future, more study on appropriate compositionality estimation model can be investigated.

Chapter 6

Event Role Level Emotion Analysis

So far, three pieces of research works are presented on different aspects of emotion analysis, namely emotion corpus construction, emotion lexicon construction, and phrase level emotion analysis. In this chapter, a more fine-grained EA task is proposed. Due to limited resources, studies on EA mainly focus on recognizing emotions expressed in a whole piece of text [105, 20]. Sometimes, the emotion expressed by an author is not necessarily linked to either the emotion of the subject or the object in the text. However, for human machine interaction, a machine needs to know the emotion state of a particular agent or a patient in a descriptive text about an event. For example , the text *"The mother hit the boy"* describes an **event**. Obviously, the emotion state of the agent *"mother"* expressed in the sentence *"The mother hit the boy"* is different from the emotion state of the agent *"mother"* in the sentence *"The mother touched the baby"*. Furthermore, the emotion states of the patient *"boy"* should also be different under the two events.

This chapter studies more fine-grained emotion analysis of agent, patient, and act in an event context. The term **event role** is used as a general term to refer to either an agent, a patient or an act in an event. Grammatically speaking, an agent and a patient in a sentence are likely to be noun phrases serving as the subject and object, respectively. The act itself is likely to be the verb of the sentence.

The research work called the Affect Control Theory (ACT) [53] and related extensions

[56] do provide a good social psychological basis. Since ACT also uses the three dimensional EPA model to represent every concept (or word), ACT is computational suited to handle emotion analysis of event roles. In ACT, every concept is represented by the multi-dimensional EPA model and every concept has a fundamental EPA representation in a specific language or culture environment. According to ACT, the EPA representation of a concept that people normally perceives without any context can change under different events. The same word *"mother"* in the previous two examples can give people different affective feelings when used in the two different contexts. Furthermore, ACT suggests that the current EPA values under a particular event, can be inferred through a regression model based on the fundamental EPA representation of a concept. One fatal drawback of ACT is that the EPA model is conceptually very difficult to understand. When all three dimensions are used to represent emotions, it is very difficult to obtain an annotated EPA lexicon with reasonable size. In other words, manual annotation is not scalable. Furthermore, using regression models cannot capture the complex semantic interaction of all the event roles involved in a particular event.

In this chapter, the Long Short-Term Memory (LSTM) network is proposed to infer the emotion of an event role in its context. The lexicon knowledge is based on automatically obtained word embeddings through unsupervised learning rather than using manually constructed EPA lexicon.

6.1 Affect Control Theory

ACT is a social psychological theory of human social interaction. ACT offers a rigorous methodology for modeling emotions in social interactions, namely events. The models can be applied to human-computer interaction leading to the design of "socially intelligent" systems that optimize user experience and outcomes [53]. In ACT, every concept is annotated under three-dimensional evaluation-potency-activity (EPA) model with the

range of [-4.3,4.3], which has been introduced for lexicon construction in Chapter 4.

Since the annotation is based on concept level words, the same word under different social environment may have different affective measures. For example, the concept *"dragon"* represents something good, powerful in Chinese while it represents something evil and powerful in English. Thus their corresponding EPA values may be different. ACT is also used in sociology to study the culture differences and EPA lexicons from different languages and culture environments are annotated separately and have proven to be indeed different. In general, within-cultural agreement about EPA meanings of social concepts is high even with consideration of across subgroups of society, and cultural-average EPA ratings from as little as a few dozen survey participants have been shown to be extremely stable over extended periods of time [54]. This means that under the same language/culture environment, the same EPA based lexicon can be used.

In ACT, every event has at least three event roles: subject (S), act or behavior (verb, V), and object (O). Each role is represented by an EPA vector. For example, *"mother"* is represented as $(2.9, 1.6, 0.5)$, *"enemy"* is represented as $(-2.1, 0.8, 0.2)$ under a common culture environment, which is called the **fundamental impression**. Fundamental impressions are those values given in an EPA lexicon. Let us use C to denote the context of an event, and the roles of the event S, V, and O are used to denote the subject, the act (which is a transitive verb to indicate the action) and object. Thus, the fundamental impression of an event can be represented as a nine-dimensional vector:

$$\vec{f_c} = [S_e, S_p, S_a, V_e, V_p, V_a, O_e, O_p, O_a], \tag{6.1}$$

where the subscripts *e*, *p*, and *a* correspond to the fundamental E, P, A values, respectively. An event can cause the emotion of a role to change from its fundamental impression to a **transient impression**, namely the context specific emotion state in an event. For example, in the event of *"The mother hit the boy"*, most readers would agree that the mother appears less nice (E), more powerful (P) and more active (A), which is the transient impression of

the subject *"mother"* in this event. An event can cause a transient impression, denoted by $\vec{\tau}$, based on the fundamental impressions of the subject, the act, and the object in the event. The transient impression of an event C can then be expressed as:

$$\vec{\tau_c} = [S'_e, S'_p, S'_a, V'_e, V'_p, V'_a, O'_e, O'_p, O'_a], \tag{6.2}$$

where each element is the transient impression of the corresponding subject, act and object. In ACT, a feature set $\vec{t_c}$ is constructed from the fundamental impression of the event as:

$$\begin{aligned} \vec{t_c} = [1, S_e, S_p, S_a, V_e, V_p, V_a, O_e, O_p, O_a, S_eV_e, \\ S_eV_p, S_eV_a, S_pV_e, S_pV_p, S_pO_a, S_aV_a, V_eO_e, \\ V_eO_p, V_pO_e, V_pO_p, V_pO_a, V_aO_e, V_aO_p, \\ S_eV_eO_e, S_eV_pO_p, S_pV_pO_p, S_pV_pO_a, S_aV_aO_a]. \end{aligned} \tag{6.3}$$

Then the ACT model obtains the transient impression $\vec{\tau_c}$ of C from the fundamental impressions of event roles by a mapping function defined by

$$\vec{\tau_c} = M\vec{t_c}, \tag{6.4}$$

where M is a parameter matrix. This is actually a linear regression model where the features of the transient impression are constructed from the fundamental impressions $\vec{f_c}$. Annotation under different languages and cultures can lead to different coefficients M. For example, the transient impression of the subject's evaluation using the US male (based on the data annotated by US males) coefficients:

$$\begin{aligned} S'_e = \text{`}.98 + .48S_e - .015S_p - .015S_a + .425V_e \\ -.069V_p - .106V_a + .055O_e + \cdots \end{aligned} \tag{6.5}$$

The learned weights can be interpreted as how much it is affected by the corresponding dimensions. The above equation shows that the transient evaluation of the subject is

mainly affected by the fundamental evaluation dimension of the subject and the evaluation dimension of the act, reflected by the positive large coefficients .48 and .425 for S_e and V_e.

Even though such kind of tasks have been widely researched in sociology, they do not attract much attention from the natural language processing community. This task can be paraphrased as: predicting the emotions of different roles in an event.

As discussed earlier, ACT is hard to scale. Even if dimensional lexicons can be extended by the proposed RoWE method in Chapter 4, dimensional affective lexicons do not show any advantage over the automatically obtained word embedding under different compositional models for phrase sentiment analysis according to the conclusion in Chapter 5. Inspired by this conclusion, a new framework is proposed to use word embedding and LSTM as the prediction model for emotion analysis of different event roles.

6.2 LSTM Based Emotion Analysis for Event Roles

Since ACT only performs prediction on events which are in the form of subject-act/behavior-object (SVO). The text used for this work are assumed to fit this pattern. Given a word sequence consists of three parts for an event C in the form of SVO, the objective is to predict the emotion of the subject, the act and the object in this event. The emotion can either be sentiment, an emotion category or multi-dimensional emotional representations such as VAD and EPA.

Inspired by the result in Chapter 5, the LSTM network is proposed for emotion prediction of different event roles using word embedding as word representations. The proposed framework is shown in **Figure 6.1** using the example sentence *"mother hit boy"*. The LSTM framework consists of three layers: the input word representation layer, the hidden LSTM layer and the output emotion layer. The input is word vector representations $\vec{x}_t$ ($t \in [1, 2, 3]$), which can either be word embedding or a multi-dimensional word affective vector. Each LSTM cell takes the current word representation $\vec{x}_t$ and the previous output

$\vec{h}_{t-1}$ of the LSTM cell as the input and outputs a hidden representation $\vec{h}_t$. The final prediction is performed on the last hidden representation (which is $\vec{h}_3$ in this case) based on the output type y. Similar to ACT, the output type y is in EPA format in this study, so the output layer beyond LSTM layer is a regression model.

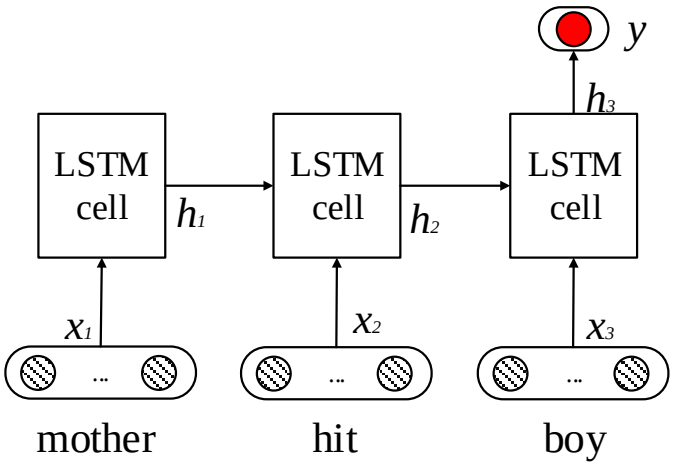


Figure 6.1: The LSTM model for emotion prediction of event roles.

For each emotion dimension (E, P, A) of each role in an event, one LSTM model is trained. The hidden dimension size is set to 4 empirically. The stochastic gradient descent method is used to train the model and the code is implemented based on Keras[1].

6.3 Experiments and Analysis

Evaluation of the proposed method is conducted using the ACT corpus from [53] as the gold answer. In this corpus, every sample consists of three words which describe an event in the form of SVO, such as *"vampire enslave heroine"*. The total size of the corpus is 515 sentences in SVO form with a vocabulary size of 106. Every word under an event is annotated with the EPA values as $\vec{\tau}$, the transient impressions. The annotation was conducted by about 25 females and 25 males of Americans using the semantic differential

[1] https://keras.io

scheme. The average of the annotations from both male and female annotators is used as the final representation. The fundamental EPA values of the 106 words are also provided by [53]. This can be used to train models such as Formula 6.4. Table 6.1 shows some examples of ACT corpus and annotated EPA values of SVO.

Table 6.1: Example samples of ACT corpus.

Sample	**Subject**			**Act**			**Object**		
	E	P	A	E	P	A	E	P	A
vampire enslave heroine	-2.31	2.0	0.73	-2.57	2.33	0.77	0.92	-1.3	1.03
daughter love baby	2.19	1.0	1.25	2.34	1.96	0.96	2.17	-1.66	2.25
daughter oppress son	-1.41	0.66	1.56	-1.42	1.16	1.45	0.48	-1.17	0.85

For word embedding, the experiment uses the available Glove 840B word embedding which is trained on a corpus of 840 billion tokens based on matrix factorization [119]. The embedding dimension is 300, denoted as **EMB**. Note that only 99 out of the 106 words in the ACT corpus appear in Glove 840B collection. To focus on the effectiveness of representation and eliminate the effect of coverage problem, only the overlap vocabulary set of the embedding and the ACT corpus is used. So, the final evaluation corpus size is actually 408 event sentences.

The performance of the predicted transient impressions of the subjects, act and object is evaluated against the gold answer set. 5-fold cross validation is performed and the best parameters obtained through manually tuning are used. The evaluation metric is the Mean Absolute Error (MAE).

The proposed method, denoted as **EMB-LSTM**, is compared with the following five baseline methods:

1. VAD-LR: This is a linear regression based method using the VAD emotion model. The features include the concatenation of the VAD values of S, V, and O. The VAD lexicon is from [167] which includes about 13K words annotated in the three dimensions of VAD. This lexicon has been introduced in Chapter 4.

2. VAD-LSTM: This is LSTM based method using the VAD lexicon as word representation. The VAD data is the same as that of VAD-LR.

3. EPA-LR: This is a linear regression based method using the EPA affective model as word representation. The EPA lexicon is provided by [53]. The features use the concatenation of the EPA values of S, V, and O.

4. EPA-LSTM: This is an LSTM based method using EPA as word representation.

5. ACT-LR: This is a linear regression based method. The input features are $\vec{t}$ defined in **Equation 6.3** constructed from the EPA values.

Feature / **Model**		VAD LR	VAD LSTM	EPA LR	EPA LSTM	ACT LR	EMB LSTM
Subject	E	.682(.059)	1.078(.039)	.556(.051)	.485(.045)	.385(.030)	**.363(.036)**
	P	.728(.031)	.751(.031)	.326(.011)	.372(.011)	**.325(.020)**	.353(.015)
	A	.539(.023)	.692(.032)	.309(.006)	.341(.029)	.313(.010)	**.274(.009)**
Act	E	.601(.093)	1.047(.229)	.467(.048)	.410(.044)	**.315(.033)**	.348(.035)
	P	.630(.053)	.637(.063)	.263(.014)	.322(.027)	.267(.014)	**.241(.024)**
	A	.496(.041)	.710(.055)	.256(.009)	.289(.011)	**.257(.009)**	.261(.041)
Object	E	.478(.030)	.561(.139)	.301(.033)	.282(.029)	.263(.036)	**.255(.035)**
	P	.791(.062)	.913(.043)	.357(.027)	.352(.037)	.355(.027)	**.277(.032)**
	A	.658(.015)	.754(.038)	.349(.020)	.318(.019)	.360(.023)	**.265(.039)**

Table 6.2: Emotion prediction of event roles based on different word representations and prediction models.

The evaluation result is shown in **Table 6.2** with the best result shown in bold. The numbers in parenthesis are standard deviations of MAE of five runs. Comparing between the manual VAD and EPA representations, EPA performs much better than VAD under both the linear regression model and the LSTM model. This is expected because the predicted emotion is represented by the EPA model in which the training data is more relevant. This, however, may also be because the VAD lexicon data is obtained through crowdsourcing, which has lower quality than the EPA lexicon. ACT features perform better than EPA

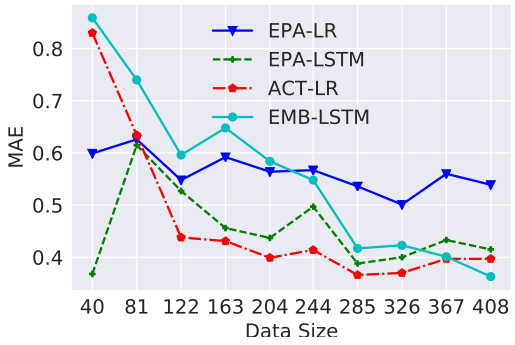
features on the evaluation (E) dimension while they perform comparatively on potency (P) and activity (A). The difference between ACT and EPA is only the additional features on the interaction of the EPA values in ACT.

Overall, the proposed EMB-LSTM has the best performance in 6 rows out of 9 as shown in bold in **Table 6.1**. ACT-LR using regression performs slightly better on three rows. One is the P dimension of the subject and the other is the E and A dimension of the act. Comparing between EPA-LSTM and EMB-LSTM, EMB-LSTM has much better performance. This clearly indicates the advantage of using word embedding than using a manually annotated lexicon. This analysis validates the effectiveness of the proposed model of using both LSTM and word embedding. Most importantly, experimental result indicates that automatically obtained word embedding outperforms manually annotated EPA and VAD lexicon for EA, which is consistent with the conclusion of Chapter 5. It should be pointed out that this experiment does not consider the coverage issue because of the limited size of the ACT corpus. If coverage issue is considered, the advantage of word embedding should be even more obvious.

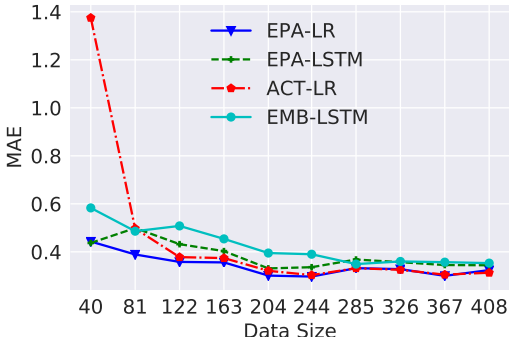
6.3.1 Effects of Data Size

Machine learning methods normally require a certain amount of training data to learn sufficient information, especially for the complex LSTM model. In the next experiment, the effects of training data size to the performance of different models are further examined. The experiment is performed on the same ACT corpus by varying the training data size starting from 40 sentences to the whole dataset of 408 sentences and run 5-fold cross validation for each affective dimension. Results in **Figure 6.2** and **Figure 6.3** show that as the training data size increases, the performance of all the models improves. However, as the data size increases, EMB-LSTM shows its better learning ability as its performance continues to improve. Overall, when the sample number reaches about 285 to 300, EMB-LSTM performs the best on most dimensions. Due to the limitation of dataset size, the

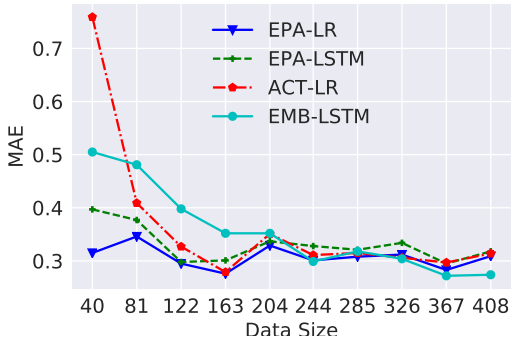
Figure 6.2: The performance on different affective dimensions of subject and act when varying the training data size.



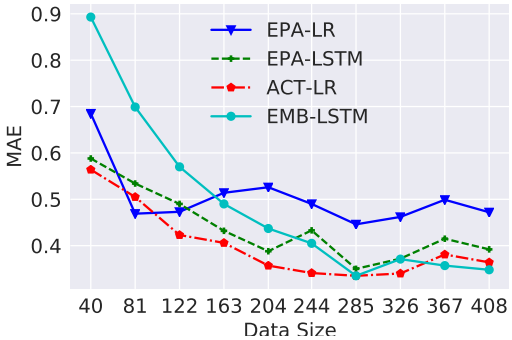
(a) E dimension of subject.



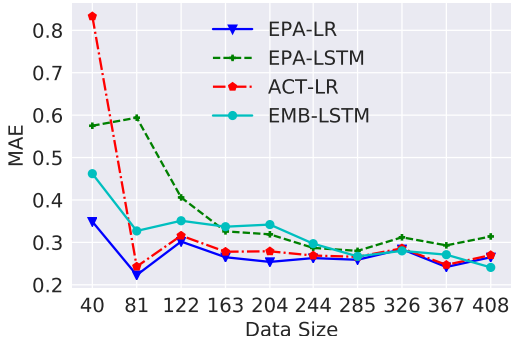
(b) P dimension of subject.



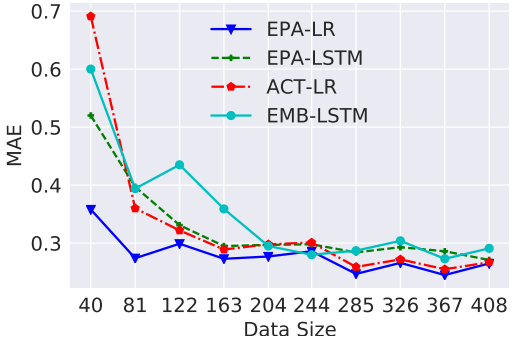
(c) A dimension of subject.



(d) E dimension of act.



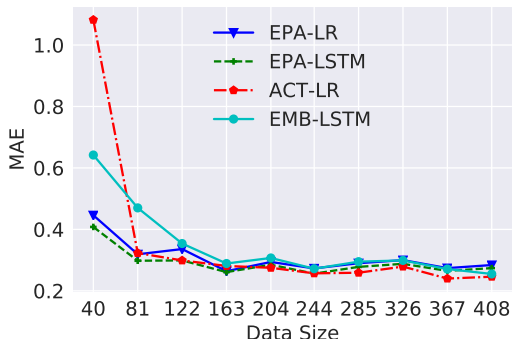
(e) P dimension of act.



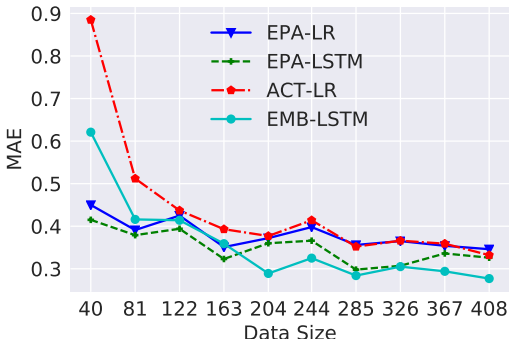
(f) A dimension of act.

performance on larger data size cannot be validated. However, the trend of performance in this experiment clearly indicates that LSTM with word embedding performs the best and

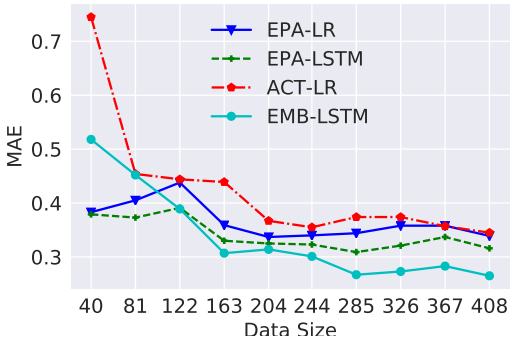
Figure 6.3: The performance on different affective dimensions of object when varying the training data size.



(a) E dimension of object.



(b) P dimension of object.



(c) A dimension of object.

more training data will likely to show even more advantages of EMB-LSTM. On the other hand, the linear regression model using EPA has very steady performance when varying the training data size. This also suggests that regression model can be useful if training data size is small.

6.3.2 Case Study

Table 6.3 lists four example events with their fundamental EPA values and their transient EPA values by different models. The column **Event** lists events in SVO form as a three word event. Column **M** shows the three models where **F** means the fundamental EPA val-

Table 6.3: Predicted transient EPA values of some example events.

Event	**M**	**Subject**			**Act**			**Object**		
		E	P	A	E	P	A	E	P	A
mother hit boy	F	1.66	1.41	-0.11	-0.8	1.2	0.9	1.06	0.29	2.13
	L	-0.01	0.16	1.13	-0.92	1.15	1.60	0.94	-0.84	1.40
	B	-0.08	1.22	0.37	-0.58	1.28	0.80	0.54	-0.52	1.53
mother touch boy	F	1.66	1.41	-0.11	1.72	0.93	0.55	1.06	0.29	2.13
	L	1.33	0.06	0.94	0.47	0.70	0.74	1.29	-0.72	1.51
	B	1.69	1.02	-0.01	1.35	0.85	0.38	1.12	0.09	1.68
teacher beat student	F	1.3	0.5	0.6	-1.17	0.78	1.39	1.49	0.18	1.87
	L	0.54	0.45	0.77	-1.36	1.11	1.04	0.76	-0.14	0.76
	B	-0.58	0.62	1.06	-1.03	0.83	1.33	0.66	-0.53	1.41
teacher teach student	F	1.3	0.5	0.6	1.6	1.1	0.9	1.49	0.18	1.87
	L	1.14	0.58	0.56	0.24	0.90	1.07	1.02	-0.65	0.79
	B	1.41	0.61	0.70	1.18	0.82	0.88	1.30	0.05	1.58

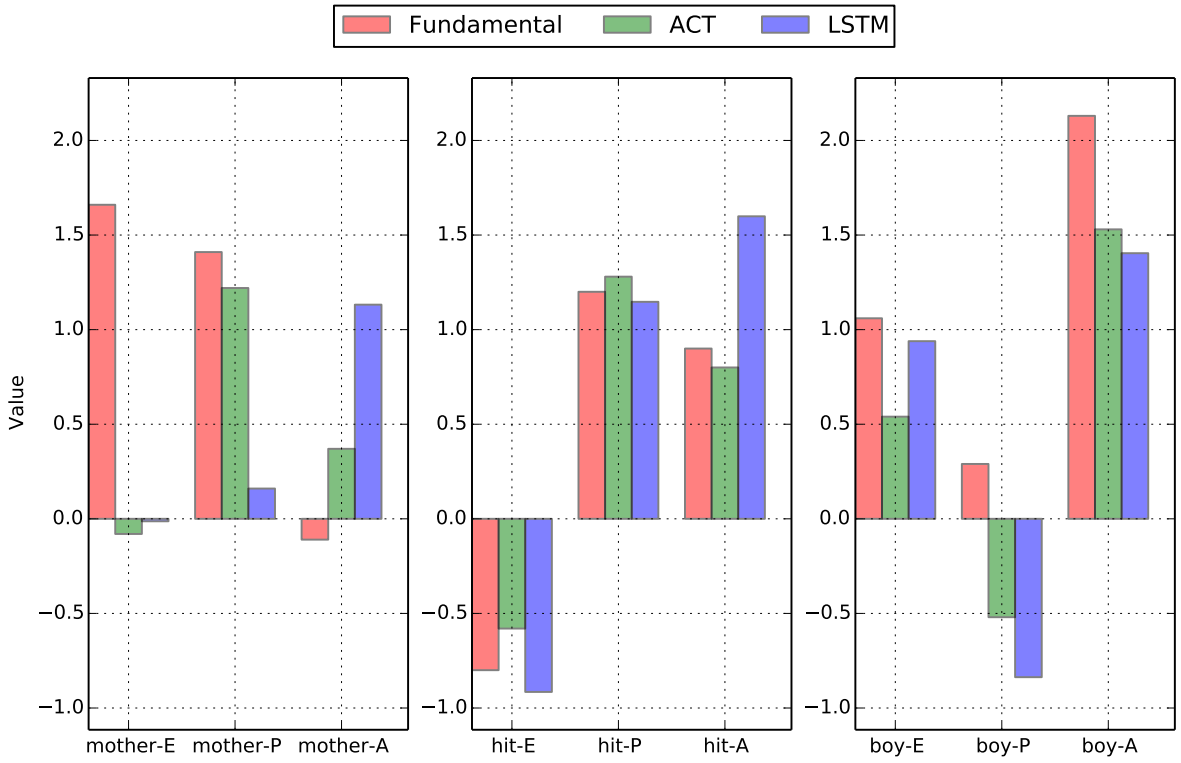


Figure 6.4: The illustration of the example SVO event *mother hit boy*.

ues, namely EPA values without context. **L** indicates the values predicted by the proposed EMB-LSTM model. **B** indicates the values predicted by the baseline ACT model. The

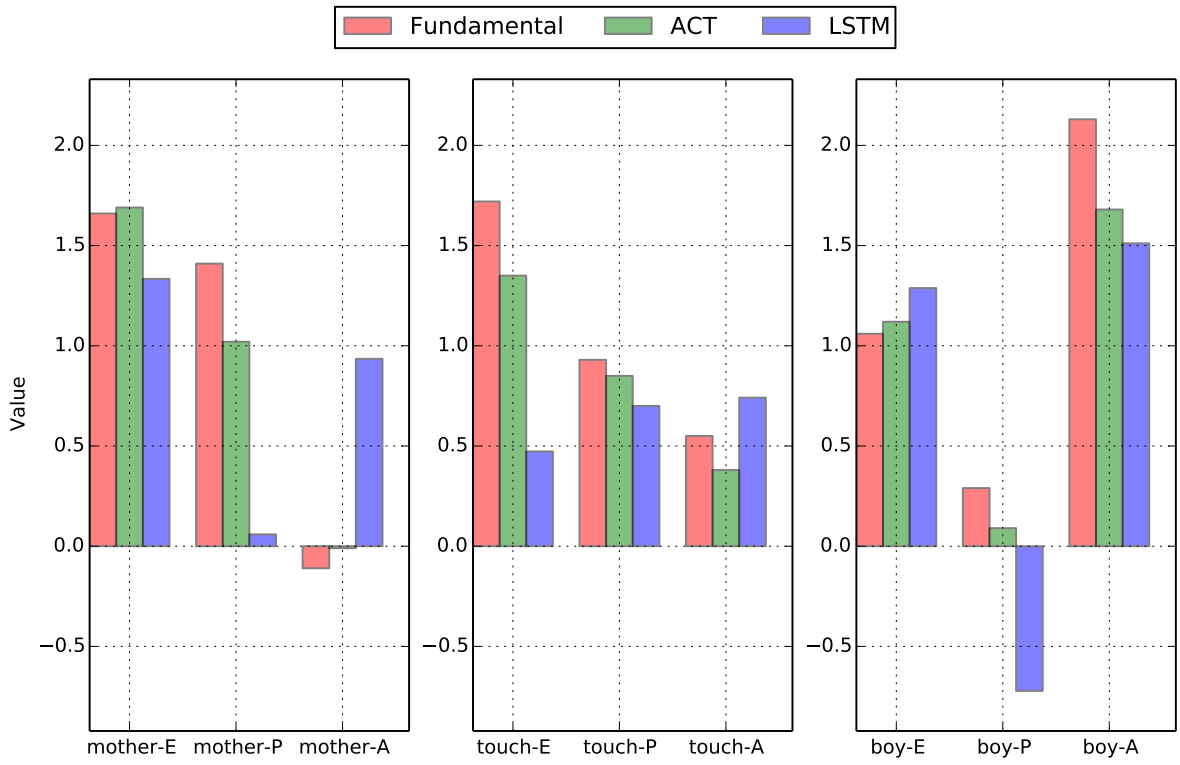


Figure 6.5: The illustration of the example SVO event *mother touch boy*.

corresponding values for the four examples are also shown from **Figure 6.4** to **Figure 6.7**, respectively, for better illustration. In **Figure 6.4** to **Figure 6.7**, the red bar shows the fundamental EPA values. The blue bar shows the values predicted by the EMP-LSTM model and the green bar shows the values predicted by the ACT model. Take the event *mother hit boy* shown in **Figure 6.4** as an example, the E value of *mother* is almost reduced to zero by EMB-LSTM. This indicates that *mother* becomes less nice under the event *mother hit boy*. Note that the P dimension of *boy* also becomes quite negative in the event predicted by EMB-LSTM. This is quite reasonable because *boy* becomes powerless in this event. The proposed model can correctly predict the transient EPA values of different event roles under a specific event.

However, EMB-LSTM may not always give reasonable predictions. For example, in **Figure 6.7** for the event *teacher teach boy*, the E value of *teach* predicted by EMB-LSTM

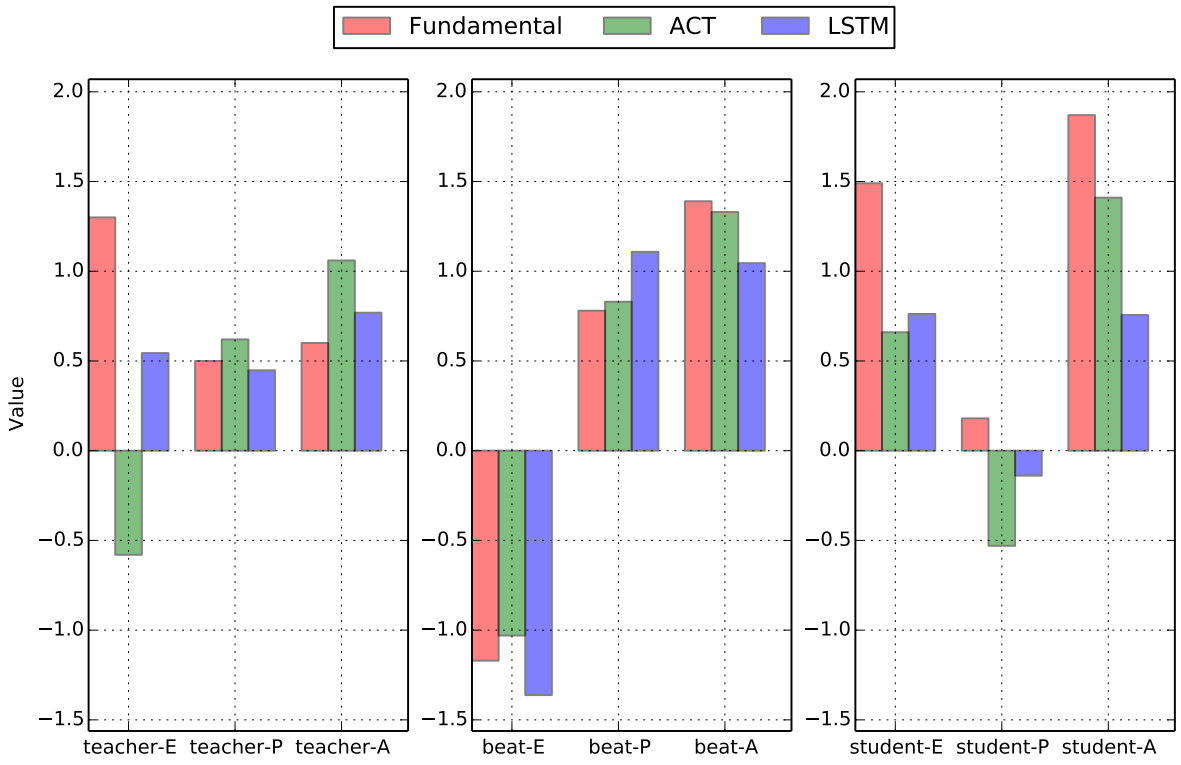


Figure 6.6: The illustration of the example SVO event *teacher beat student*.

is smaller than the fundamental E value, which contradicts common sense. In such an event, the evaluation of *teach* should be nicer. This may be the result of under-training by the LSTM model and the availability of more training data may give more insight to this result in the future.

6.4 Chapter Summary

In this chapter, a novel emotion analysis task is proposed to predict the emotions of different event roles, including subject, act and object involved in a described event, which is inspired by the research in sociology. The emotions are represented using three-dimensional EPA model instead of using discrete sentiment or emotion labels. The main idea of the proposed approach is to use automatically obtained word embedding as word representation and to use the Long Short-Term Memory network as the prediction model. Performance

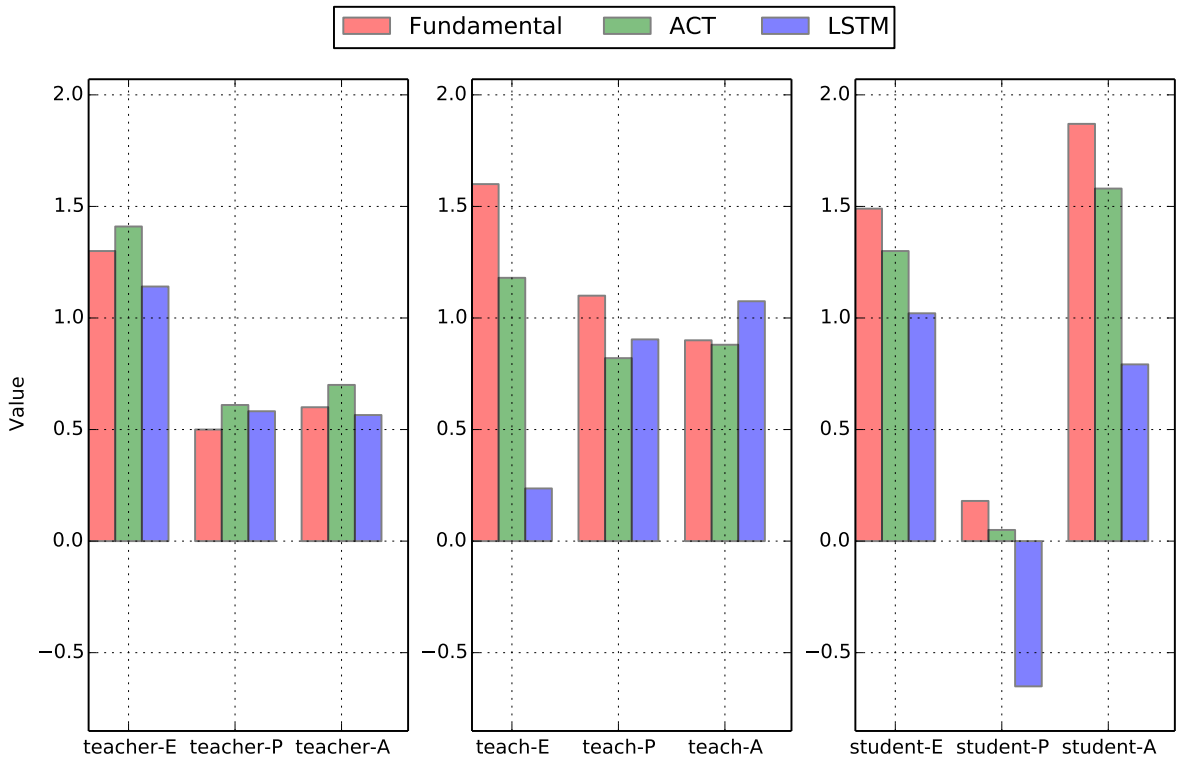


Figure 6.7: The illustration of the example SVO event *teacher teach student*.

evaluation is conducted using an event corpus with each subject, act and object annotated with EPA values under event context. Compared to the linear model used in ACT which uses a manually annotated EPA lexicon, the proposed LSTM with word embedding outperforms the linear model on most affective dimensions. Most importantly, the result indicates that automatically obtained word embedding outperforms manually constructed affective lexicons. In fact, the experiments indicate that a seemingly sound model in theory, may not work well computationally. One limitation of this study is the limited corpus size and one future direction can be on how to build such kind of corpus in large scale. Another limitation of current work is that the proposed model can only handle structured sequence with the form of subject-act-object. Additional information in sentences such as time, place and other descriptive features are not being used. Extending the proposed model to more complex event descriptions will be a future direction.

Chapter 7

Conclusions and Future Work

Emotion analysis is a very important research area in artificial intelligence to enhance computer systems, as well as autonomous agents and robots to recognize and express emotions like human beings. One of the key elements in this area of research is the understanding of emotions expressed in text. This thesis studies emotion analysis from text. The focus is on how to use emotion models defined in dimensional space for emotion prediction at different levels of text units. This chapter reserves to conclude this thesis. The main contributions of this thesis will be summarized first, followed by limitations and future work.

7.1 Contributions

This thesis covers a comprehensive range of studies on emotion analysis from emotion corpus construction, emotion lexicon construction to emotion analysis model. The main conclusions and contributions are summarized as follows:

1. **Emotion corpus construction.**

   A general framework is proposed to automatically or semi-automatically filter out noisy data in a naturally annotated corpus. This framework enhances the current distant supervision based corpus acquisition method by introducing two additional steps to automatically identify high-quality data, followed by an optional manual

step to obtain more data further. Based on this proposed framework, a high-quality emotion corpus of size 43,300 for Chinese social network data of microblogs are made available for public access. This framework can also be used to build other corpora that are based on distant supervision method.

2. **Dimensional emotion lexicon construction.**

   This work proposes an effective method to build dimensional emotion lexicons in large scale. The proposed method uses a regression model to infer the affective meanings of words from word embeddings learned in the general domain. This method achieves the state-of-the-art result on inferring different affective meanings under different emotion models, including the sentiment, Valence-Arousal-Dominance (VAD), Evaluation-Potency-Activity (EPA), Evaluation-Activation-Imagery (EAI). The method is not only proven to be effective in prediction of affective meanings but also effective in other meaning dimensions. In addition, this method also has computation advantage over the other baseline methods. As a result of this work, several extended multi-dimensional emotion lexicons with size of million scale are made available publicly. In addition, experiments on different word embeddings show that incorporating multiple semantic evidence, such as knowledge base, can lead to better word embedding and better predictions of affective meanings.

3. **Phrase level emotion analysis.** This part consists of two pieces of work. The first work explores effective word representations to be applied to compositional models to obtain affective representation of phrases. Results indicate that domain specific dimensional emotion representations do not have an advantage over representations using word embeddings although domain specific affective representations are based on sound psychological foundations. This indicates that automatically obtained word embeddings which encode more semantic information, are more suitable for compositional models to obtain the representation of phrases. Affective

meanings of words alone cannot capture all semantic information during the composition process to infer the affective meanings of larger text units. As we know, compositional methods can fail when a phrase is non-compositional. Thus, in the second work, I attempt to directly learn embedding representations of phrases. A new phrase embedding learning model is proposed by taking into consideration of both the distributional hypothesis and the principle of compositionality. The proposed hybrid model combines the distributional component, which takes care of the external context, with the context of the component words with a compositionality constraint. Evaluation on four phrase semantic tasks shows that the proposed model is more robust than both the compositional methods and the distributional methods. The directly learned phrase embedding performs much better than the baselines on a phrase sentiment prediction task.

4. **Fine-grained emotion analysis for event roles.**

   A novel emotion analysis task is proposed to predict the emotions at fine-grained level. Rather than focusing on emotions expressed by a whole piece of text, this task aims to predict the emotions of the subject and object involved in an event in the form of subject-verb-object. Thus, we refer to it as emotion analysis at the event role level. The main idea is to use automatically obtained word embedding as word representation and use the LSTM as the prediction model. Most importantly, separate LSTM networks are built for different event roles and different affective meanings. Compared to the linear model used in Affect Control Theory (ACT) from sociology which uses a manually annotated EPA lexicon, the proposed LSTM with word embedding outperforms the linear model on most affective dimensions. This novel task focuses on fine-grained event role level emotion prediction, which has many potential applications. For instance, in an emotion-aware dialog system, this work can help to produce responses based on the emotions of the event roles

involved in the conversation.

7.2 Limitations and Future Work

Some limitations of this thesis are summarized here. Firstly, in emotion corpus construction, the proposed framework does not use samples that have no naturally annotated emotion labels such as hashtags to indicate its affective meaning. Also, the lack of affect-linked hashtags does not imply the sample is neutral. Thus, even though the obtained emotion corpus is much larger than other resources available, the corpus does not contain neutral samples. Secondly, for multi-dimensional affective lexicons construction, the current work cannot distinguish the words that have multiple senses. This results from the fact that word embedding cannot distinguish different word senses effectively. The scope of this thesis is under the assumption that each lexical word is associated with one emotion label (or one set of values in a multi-dimensional space. However, some words may be associated with mixed emotion labels. For example, *"tragicomedy"*, can be associated with both *sadness* emotion or *happiness* emotion. The same is true in Chinese, such as the Chinese word *"*悲喜交加*(mixed feelings of grief and joy)",* which have emotions of *sadness* and *happiness*. This cannot be directly predicted from a single word embedding. Thirdly, for phrase embedding learning, the proposed method requires pre-computed compositionality of the phrases and the performance relies on the accuracy of the compositionality measures of the phrases. The current method for calculating compositionality is not quite accurate, and thus the performance of the proposed D&C method still has room for improvement. Fourthly, for fine-grained emotion analysis of event roles, the gold answer has only 480 samples, which is too small under the framework of deep learning. Such small dataset cannot make full use of the complex models that are data eager, such as LSTM.

Future directions include: 1) Exploring methods to include neutral samples into annotated emotion corpus. One of the possible solutions may crawl the microblogs that do

not have naturally annotated labels and employ classifier to classify those samples. Those with low classification confidence can be considered as the candidates of neutral samples. 2) Obtain better word embedding, including word embedding for multiple senses. The work in Chapter 4 indicates that incorporating multiple semantic evidences can lead to better word embeddings. One future direction is to explore ways to incorporate different semantic evidences to learn better word embeddings. The semantic evidences can include knowledge base, images of concepts, concept definitions, morphemes of concepts, etc. 3) Incorporate common sense reasoning in the composition process to learn representations of larger text units. When people read a sentence, common sense reasoning is naturally used by us for simultaneous associations. If this can be incorporated, we would be one step closer to true intelligence by computers. 4) Explore better methods for estimating compositionality. One potential direction is to merge the compositionality prediction process into the embedding learning model and learn compositionality and the phrase embedding simultaneously as done in [50].

Appendices

Appendix A

Samples of the Annotated Emotion Corpus Using the 6 Step Approach

The following table lists some samples of the emotion corpus constructed using the 6 step approach given in Chapter 3. The complete corpus can be downloaded.[1]

Table A.1: Samples of built emotion corpus.

Label	**Text**
sadness	经历了一些事情，你就更不愿意把悲伤说给别人听。
sadness	你说最害怕女孩子哭，所以我偷偷地在你不在的时候轻轻哭着，低着头将声音压到最低，就算身边再多人也不会有人发现，让你看见我哭了好几次，对不起，我真的没忍住。
sadness	可以触摸的痛苦是什么？就是觉得肚子都饿扁了，一摸还是有一坨肉。
sadness	住院的日子真难熬！
sadness	经历了一些事情，你就更不愿意把悲伤说给别人听。
sadness	无端端又被shoot
sadness	雨一直下，人一直在
sadness	昨天乐妈辛苦写好的一篇微博，放在浏览器里没来及发布，就被乐爸玩手机时关闭了浏览器，全没了，白写了！！哭！！
sadness	为啥么我一买泡面就买到叉子活动不结实的？？？？ [抓狂][抓狂]神啊！敢告诉我是神马情况不？？？？？ [抓狂]
sadness	疯了！买个蛋糕的功夫车竟然打不着了！作啊～
sadness	其实我也是个很自私，很小心眼的人。有些事，有些话，你做给我了说给我了就别再给别人。
sadness	至始至终，只能把对你的相思寄托，只有寄托才能让我的内心不再寂寞。
sadness	事实上我也蛮佩服自己的笑点，同一个梗我大概听上几千遍都还会笑，并且还会在心里想怎么这么好笑啦！！

[1] https://yunfeilongpoly.github.io/Team_resource.html

sadness	早上泡牛奶，找很久都没看不到亲耐滴勺子，突然想起昨晚吃宵夜后好象把它和饭盒一块打包扔了…
sadness	=_=。不越狱真是…除了输入法其他什么都能下…
sadness	妈一口喝完了我花了两小时纯手工榨的雪梨汁！[bm哭诉]
sadness	[bm抓狂]下周忙死的节奏，好不容易逮到机会可以去南航玩，结果发现那天被抽到体测，我去，让不让人活了！我只想放松下，我容易吗！[泪流满面]
sadness	我还是忘不了你。
like	第一天，晚安～
like	我想和你度过每一个晨昏。
like	我于千万人之中，见过你的发，你的眼，却始终不是你的脸。亲爱的少年，原谅我没有勇气走到你身边。不过这样也好，我可以有很多的时间来想念你，就算你不在，我还是爱。
like	开始筹备创业的事情，愿上帝带领我，让我能在工作中去荣耀他的名。
like	那种友情，那种亲密，那种什么什么，都让我不可及。
like	“赞”这个字的实际含义已经被各类社交平台剥离它本身，肢解得体无完肤，延伸的个中意义耐人寻味，暧昧得不可仔细琢磨，常用来表达含糊不清的态度，这一手法被广泛使用逐渐形成一个强大的组织——点赞党！
like	元丰六年十月十二日夜，解衣欲睡，月色入户，欣然起行。念无与为乐者，遂至承天寺，寻张怀民。怀民亦未寝，相与步于中庭。庭下如积水空明，水中藻荇交横，盖竹柏影也。何夜无月？何处无竹柏？但少闲人如吾两人者耳。（930年前的夜。心平，景美。930年后的今夜，四一很想吃烧烤…）
like	妈，你怎么这么可爱嘞，哇哈哈哈哈哈哈哈哈哈[哈哈]
like	每天都在做作业啊……还有额外任务……奋斗吧，坚持一下就好[赞][呵呵]
like	不知道你梦见的会不会是我、哈哈
happiness	所谓猪一样的室友，应该就是我感冒了，让他回来给我带一盒白加黑，他给我带了一包奥利奥。
happiness	点点滴滴。特别是来自部长，sosweet！2014-819
happiness	钟芮在昆明买给我的
happiness	首页上怎么全是汉化本子啊，大家都约好了发的么，都连下5、6本了……
happiness	就是刚想早睡却突然发现明天是周二不用上班 [嘻嘻]
happiness	老公，今晚做的饭好香啊。馋嘴…
happiness	“你唱歌的声音最迷人”[亲亲]
happiness	聊了一个小时，最后感动一句话：君以国士待我，我当以国士报之！
happiness	周末加油站第一站幸福大讲堂接近尾声！王老师寄语：用心爱身体，每天做好四件事：1、吃饭；2、睡觉；3、工作；4、锻炼。幸福的载体是你自己，想要幸福先得健康！感谢全体参加TCL大讲堂的同事
happiness	今天是元宵节，虽不能和家人一起过，但朋友为了我也没有回家和家人过，请我吃饭，聊天，看韩剧。原来自己从不孤独。身边时刻会有人陪着。[爱你]

happiness	这个秋天，这个国庆，苹果小超人来袭，准备好了嘛，准备好一起和我们一起飞啦嘛[可爱][可爱][太开心][太开心]
happiness	快乐无处不在，只要你每天保持好的心态、好的心情去过好每一天，你的生活会变得很美好，每天给予自己一个微笑，告诉自己前方是一片光明的。
happiness	[语录]人活着无非是一种状态，如果不能去做自己喜欢的是而留下遗憾，这就是老，心理的老比身体的老更可怕。[阳光]早上好
happiness	就是，坚持了应该坚持的，放弃了应该放弃的，珍惜现在拥有的，不后悔已经决定的。
happiness	被一个笑话戳中笑点：养的小仓鼠生病了，不过没关系，家里有老鼠药，希望它吃过以后能好起来。恩恩恩！
happiness	友人问我为何你每天开心，好像没有忧愁。我答，我想与别人不一样。
happiness	飞得更高现场版，赞！！！
disgust	爆吧有何意义？一群神经病。
disgust	看完金粉世家看奋斗，下一步是不是只有甄传了[懒得理你][懒得理你]
disgust	微博越来越没有意思了有木有！！！！！！！
anger	取钱光排队排一个多小时，前面的人都不动！！！！
anger	动力什么的都去死吧，看半个小时的书，居然睡2个小时。死的惨也都是自己活该[蜡烛]
anger	某某某，某某，你们再不还钱，我真的要拖家带口去你们家过年了，有意思不，有意思不！！！！！！
anger	刚保安阿姨冲上图书馆四楼，对着对讲机大喊一声‘没有！’震惊全场，把人吓尿，随后踩着将近10厘米的高跟鞋跑走了
anger	说起手头这个作孽的项目，大暴斯说“一定要把这个大腿抱住!”
anger	[衰]久久遇着堵堵真是跳进黄河也洗不清无语……[睡觉]
anger	昨天晚上做梦梦到把贞子好一顿调戏。。。
anger	我草泥马，身体给我好起来啊！今天同学聚会给我撑过去啊！
anger	[怒][怒]死了这么多人，家里的妻儿老小他们不哭吗[泪]。为了祖国的建设不要哭[泪][泪]。
anger	新浪你敢不敢在我每次登陆网页版的时候不给我推荐那些逗B让我关注?
surprise	饭后睡前宜有氧运动，不宜看球造成情绪波动。
surprise	看到伐，说有一种软件是预约出租车的，就是和出租车司机谈价钱，比如要到某地去你愿意比原价多出50元，司机与你一拍即合的话，那就预约成功。
surprise	奇怪。。。怎么发微博输入一个微话题马上吸引两位僵尸粉。。。
surprise	【大四女生丢钱包后与小偷对视小偷不停擦汗最终还包】西安一高校的大四女生小崔在公交上，发现钱包没了，然后她看到旁边一年轻男子正紧张的看着自己，感觉他是小偷，于是直直地盯着他的眼睛，对方被她盯的不停擦汗、咽口水，一分钟后，小崔叹了口气伸出手，小偷就把钱包还给她了[吃惊]
surprise	拜托，我有女朋友了好不好，还是个射手座的，要是被她知道你这样，我该怎么说啊，射手座那么不好哄，你还是走远点吧
fear	我突然觉得有点怕爱跟生活的一切

Appendix B

Examples of Extended Multi-dimensional Lexicons

The following tables list sampled words[1] of the extended multi-dimensional lexicons in **Chapter 4** based on the CVNE word embedding (except for the Chinese CVAW lexicon which is based on the word embedding learned from Baidu Baike corpus). In each table, the samples are selected by top, middle and bottom n words in each affective dimension based on the predicted values. For example, in **Table B.1**, words from number 1 to 5 all have high valence values, words from number 6 to 10 all have middle valence values and words from 11 to 15 all have low valence values. Subsequent tables follow the same pattern. The complete lexicons based on different word embeddings can be downloaded.[2]

Table B.1: Examples of extended ANEW lexicon (dimensions of Valence-Arousal-Dominance) based on CVNE word embedding.

Num	Word	Valence	Arousal	Dominance
1	happiness	9.13	5.86	6.62
2	enjoy	9.17	5.61	6.77
3	enjoying	9.19	5.61	6.68
4	felicific	9.35	5.34	6.83
5	gifts	9.35	6.64	6.71
6	reattend	4.74	4.92	4.68

[1] CVNE also contains many phrases because CVNE is based on ConceptNet, which contains many phrase level concepts. Here only single words are selected.

[2] `https://yunfeilongpoly.github.io/Team_resource.html`

7	physiographer	4.74	4.65	4.34
8	aberginian	4.74	5.15	5.15
9	crawfordite	4.74	4.71	4.68
10	brumously	4.74	3.97	4.5
11	plague	0.21	5.55	3.22
12	plaguer	0.24	5.4	3.2
13	hagridden	0.49	6.77	3.07
14	parasitophobia	0.51	6.03	3.15
15	thanatophobia	0.51	6.54	2.74
16	enraged	2.46	7.97	6.33
17	thrill	8.05	8.02	6.54
18	rollercoaster	8.02	8.06	5.1
19	orgasm	8.32	8.1	6.83
20	rage	2.41	8.17	5.68
21	incorruptibly	5.36	4.76	4.77
22	corporosity	5.49	4.76	5.1
23	asynchronously	3.85	4.76	4.13
24	cuzco	5.23	4.76	4.79
25	adenodiastasis	3.01	4.76	3.93
26	relaxed	7.0	2.39	5.55
27	paper	5.2	2.5	4.47
28	unfigured	4.81	2.61	4.47
29	fatigued	3.28	2.64	3.78
30	footstall	4.41	2.64	4.67
31	king	7.26	5.51	7.38
32	win	8.38	7.72	7.39
33	admired	7.74	6.11	7.53
34	confident	7.98	6.22	7.68
35	leader	7.63	6.27	7.88
36	postcoded	4.31	4.42	4.65
37	medifixed	4.84	3.53	4.65
38	pleck	4.19	5.49	4.65
39	nicad	4.63	4.2	4.65
40	accuminate	4.28	3.68	4.65
41	helpless	2.2	5.34	2.27
42	insecure	2.36	5.56	2.33
43	failure	1.7	4.95	2.4
44	indisposing	0.85	5.22	2.51
45	loneliness	1.61	4.56	2.51

Table B.2: Examples of extended CVAW (dimensions of Valence-Arousal, Chinese) lexicon based on Baidu Baike word embedding.

Num	Word	Valence	Arousal
1	狂喜	8.6	8.8
2	尚美	8.63	4.11
3	品尚	8.65	5.0
4	同辉	8.69	5.98
5	大风车	8.72	5.47
6	预祝	8.83	5.89
7	共绘	9.04	5.55
8	万事如意	8.58	5.52
9	操碎了心	4.36	6.14
10	连接轴	4.36	4.99
11	邀您	8.6	6.12
12	通道式	4.36	5.44
13	青伊湖	4.36	6.41
14	挖眼	0.82	7.62
15	刑讯	0.86	7.16
16	株连	0.89	7.83
17	逼供	0.91	7.78
18	非法拘禁	0.92	7.29
19	弑君	0.94	7.81
20	狂暴	1.8	8.8
21	狂喜	8.6	8.8
22	怒骂	1.8	8.8
23	怒吼	2.0	8.8
24	干	1.0	8.8
25	热血沸腾	5.12	8.82
26	狂潮	4.8	8.94
27	寻来寻	4.03	5.94
28	前十	5.5	5.94
29	创味	4.46	5.94
30	寒从脚下起	3.38	5.94
31	酷客	6.71	5.94
32	郭家崖	4.31	5.94
33	宁静	6.2	1.6
34	镇静	5.4	1.8
35	放松	6.2	2.0
36	闲散	4.6	2.2
37	轻松	6.0	2.2

Table B.3: Examples of extended EPA (dimensions of Evaluation-Potency-Activity) lexicon based on CVNE word embedding.

Num	Word	Evaluation	Potency	Activity
1	saint	3.15	2.22	-0.3
2	honeymoon	3.22	2.05	1.49
3	angel	3.3	2.22	0.59
4	blessings	3.35	1.65	0.12
5	heaven	3.49	3.01	-0.5
6	circumforanean	0.28	-0.44	0.28
7	cybernationalism	0.28	0.43	0.95
8	brassart	0.28	0.87	0.28
9	chinesely	0.28	0.29	0.31
10	rapist	-3.94	-0.22	0.59
11	rape	-3.53	0.69	1.55
12	murder	-3.51	0.86	1.07
13	hell	-3.49	1.95	1.12
14	heaven	3.49	3.01	-0.5
15	pope	2.85	3.05	-1.62
16	christ	2.81	3.14	0.57
17	ceo	0.63	3.16	-0.56
18	god	2.97	3.34	0.07
19	scrotum	-0.39	0.32	0.1
20	aulonemia	0.64	0.32	0.37
21	felts	0.64	0.32	-0.01
22	ethoxybutamoxane	-0.54	0.32	0.62
23	powerless	-1.85	-2.7	-0.99
24	slave	-0.4	-2.3	-0.19
25	coward	-1.14	-2.29	-0.63
26	weakling	-0.43	-2.29	-0.85
27	nightclub	1.6	1.37	2.68
28	fighter	-0.51	2.29	2.75
29	gunfight	-2.92	1.86	2.81
30	riot	-1.93	2.27	2.83
31	raver	0.65	-0.54	3.08
32	oxidopamine	0.13	0.33	0.38
33	echinococcosis	-0.36	0.68	0.38
34	ardea	1.6	0.78	0.38
35	contemporary	1.62	0.86	0.38
36	graveyard	-0.87	0.14	-2.68
37	mummy	-1.19	1.0	-2.4

Table B.4: Examples of extended DAL (dimensions of Evaluation-Activity-Imagery) lexicon based on CVNE word embedding.

Num	Word	Evaluation	Activity	Imagery
1	beautifully	3.0	1.33	2.0
2	softly	3.0	2.25	1.0
3	happyness	3.01	2.25	2.12
4	lovewende	3.01	2.07	1.84
5	happines	3.08	2.52	2.16
6	allosteric	1.69	1.76	1.51
7	sayer	1.69	1.86	1.53
8	unrug	1.69	1.68	2.05
9	accelerationist	1.69	2.13	1.28
10	plaguer	0.61	2.06	1.72
11	nidder	0.61	2.19	2.24
12	plague	0.63	2.0	2.02
13	mommick	0.67	1.57	1.49
14	arrested	1.0	3.0	2.4
15	energy	2.0	3.0	2.4
16	victor	2.5	3.0	2.0
17	speed	1.83	3.0	1.6
18	travel	2.57	3.0	1.6
19	rereinforce	1.98	1.8	1.14
20	stenopelmatidae	1.65	1.8	2.17
21	mavens	1.62	1.8	1.54
22	lakesha	1.72	1.8	1.69
23	oxgang	1.72	0.99	2.07
24	unconscious	1.38	1.0	2.2
25	mm	1.8	1.0	1.4
26	housed	2.0	1.0	1.6
27	heraldiccharge	1.63	1.27	3.36
28	kitten	2.18	1.95	3.42
29	skibob	2.04	2.12	3.45
30	sandboard	2.04	2.13	3.49
31	petshop	2.1	1.97	3.52
32	nonclient	1.86	1.87	1.75
33	gathers	1.89	2.02	1.75
34	prediastolic	1.98	1.83	1.75
35	ritters	1.84	1.94	1.75
36	inhere	1.71	1.6	0.12
37	risibility	1.92	1.59	0.15

Table B.5: Examples of extended VADER lexicon (dimension of Sentiment) based on CVNE word embedding.

Num	Word	Sentiment
1	superfabulous	3.34
2	wealful	3.35
3	douth	3.36
4	gustoso	3.37
5	excellenter	3.37
6	resplend	3.37
7	ily	3.4
8	magnificently	3.4
9	concinnity	3.46
10	snazztastic	3.47
11	goodful	3.51
12	felicitations	3.55
13	excellentness	3.73
14	confuciusornithid	0.1
15	superoperon	0.1
16	pressurizer	0.1
17	groundation	0.1
18	bryanthus	0.1
19	dargin	0.1
20	glyoxysome	0.1
21	sedation	0.1
22	jamil	0.1
23	polymignyte	0.1
24	splurges	0.1
25	velverd	0.1
26	hagride	-4.25
27	hagridden	-4.09
28	rapist	-3.9
29	parasitophobia	-3.82
30	slavery	-3.8
31	raping	-3.8
32	nithing	-3.8
33	crybully	-3.78
34	necrophobia	-3.77
35	plague	-3.75
36	rape	-3.7
37	kill	-3.7

Table B.6: Examples of extended Perceptual lexicon (dimensions of Hearing-Tasting-Touching-Smelling-Seeing) based on CVNE word embedding.

Num	Word	Hearing	Tasting	Touching	Smelling	Seeing
1	noises	5.77	0.52	0.73	0.98	2.17
2	heard	5.85	1.06	0.64	0.76	1.77
3	shouts	5.98	-0.03	0.36	0.31	2.89
4	devolatilizer	1.65	1.57	1.85	1.55	3.35
5	simolean	1.65	0.61	1.36	0.9	2.99
6	gules	-1.47	0.73	0.37	0.25	3.95
7	torteau	-1.34	0.83	1.15	0.44	4.06
8	saporous	0.38	5.96	0.96	4.76	2.3
9	sipid	0.22	5.97	0.88	4.39	2.11
10	savorsome	0.29	5.97	0.96	4.45	2.5
11	reebless	0.92	0.93	2.81	0.67	3.55
12	laune	1.13	0.93	0.74	1.38	3.38
13	decameter	1.45	-1.16	1.47	-0.41	3.71
14	petavolt	1.85	-1.14	1.25	-0.72	3.55
15	calloused	1.52	0.87	5.42	0.2	3.88
16	callused	1.48	0.43	5.69	0.19	3.85
17	wristwarmer	0.52	-0.12	5.94	0.62	4.65
18	nonreligious	2.13	1.03	1.61	0.71	3.0
19	inobedient	2.48	1.09	1.61	0.69	3.04
20	nox	1.38	0.38	-1.13	1.59	2.63
21	millilux	0.94	-0.39	-1.08	0.86	2.64
22	kukumakranka	0.11	4.3	1.88	5.46	2.96
23	empyreuma	1.31	4.11	2.46	5.55	3.3
24	smells	1.32	3.28	0.79	5.62	1.56
25	bullier	2.71	0.89	1.21	1.07	3.38
26	subadult	1.71	1.19	2.14	1.07	4.12
27	aposiopesis	2.95	-0.84	0.39	-1.17	2.4
28	cataphora	2.7	-0.64	-0.34	-1.07	2.26
29	optigraph	-0.1	-0.03	2.59	0.2	5.58
30	eumelanic	0.4	0.26	2.67	0.84	5.58
31	oroheliograph	0.22	0.07	2.03	0.29	5.61
32	groupe	1.88	0.83	1.13	1.09	3.4
33	acclimates	2.09	1.21	2.1	1.55	3.4
34	perfumed	0.1	2.29	0.19	4.9	0.48
35	echoing	4.71	0.0	0.33	0.0	0.52

Table B.7: Examples of extended Concreteness lexicon (dimension of Concreteness) based on CVNE word embedding.

Num	Word	Concreteness
1	landsailor	5.58
2	refridgerator	5.59
3	chamfron	5.59
4	gugelhupf	5.6
5	fingerstall	5.6
6	hallstand	5.6
7	alvus	5.62
8	topek	5.63
9	pileable	5.63
10	vesre	5.67
11	skibob	5.77
12	petshop	5.8
13	heraldiccharge	6.12
14	streisand	2.99
15	dihydroquinoline	2.99
16	aurist	2.99
17	respins	2.99
18	proteobacterium	2.99
19	unserdeutsch	2.99
20	endura	2.99
21	thuris	2.99
22	gynecologists	2.99
23	euronesian	2.99
24	defects	2.99
25	bandera	2.99
26	istically	0.35
27	hypostatize	0.51
28	confessedly	0.52
29	undownable	0.63
30	affectual	0.63
31	hypostatise	0.65
32	ostensively	0.66
33	infelicitously	0.67
34	apodeictic	0.67
35	declaredly	0.7
36	affectioned	0.75
37	superlation	0.76

Bibliography

[1] Apoorv Agarwal, Fadi Biadsy, and Kathleen R. Mckeown. Contextual phrase-level polarity analysis using lexical affect scoring and syntactic n-grams. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 24–32. Association for Computational Linguistics, 2009.

[2] Areej Alhothali and Jesse Hoey. Good News or Bad News: Using Affect Control Theory to Analyze Readers' Reaction Towards News Articles. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*, pages 1548–1558, Denver, Colorado, USA, 2015.

[3] Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. Emotions from Text: Machine Learning for Text-based Emotion Prediction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 579–586, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[4] Ebba Cecilia Ovesdotter Alm. *Affect in Text and Speech*. VDM Verlag Dr. Müller, 2009.

[5] Saima Aman and Stan Szpakowicz. Identifying expressions of emotion in text. pages 196–205. Springer, 2007.

[6] Silvio Amir, Ramón Astudillo, Wang Ling, Bruno Martins, Mario J. Silva, and Isabel Trancoso. INESC-ID: A Regression Model for Large Scale Twitter Sentiment Lexicon Induction. In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT*, pages 613–618, Denver, Colorado, USA, 2015. Association for Computational Linguistics.

[7] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, volume 10, pages 2200–2204, 2010.

[8] Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. *From Tweets to Polls : Linking Text Sentiment to Public Opinion Time Series*. 2010.

[9] Anil Bandhakavi, Nirmalie Wiratunga, Deepak P, and Stewart Massie. Generating a Word-Emotion Lexicon from #Emotional Tweets. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics (*SEM 2014)*, pages 12–21, 2014. 00000.

[10] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland, 2014. Association for Computational Linguistics.

[11] Marco Baroni and Roberto Zamparelli. Nouns are Vectors, Adjectives are Matrices: Representing Adjective-Noun Constructions in Semantic Space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1193, Cambridge, MA, 2010.

[12] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A Neural Probabilistic Language Model. In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, pages 932–938, 2000.

[13] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.

[14] Chris Biemann and Eugenie Giesbrecht. Distributional Semantics and Compositionality 2011: Shared Task Description and Results. In *Proceedings of the Workshop on Distributional Semantics and Compositionality*, DiSCo '11, pages 21–28, Stroudsburg, PA, USA, 2011.

[15] Sasha Blair-Goldensohn, Tyler Neylon, Kerry Hannan, George A. Reis, Ryan Mcdonald, and Jeff Reynar. Building a sentiment summarizer for local service reviews. In *In NLP in the Information Explosion Era*, 2008.

[16] Roger Bougie, Rik Pieters, and Marcel Zeelenberg. Angry customers don't come back, they get back: the experience and behavioral implications of anger and dissatisfaction in services. *Journal of the Academy of Marketing Science*, 31(4):377–393, 2003.

[17] Margaret M Bradley and Peter J Lang. Affective norms for English words (ANEW): Instruction manual and affective ratings. Technical report, Technical report C-1, the center for research in psychophysiology, University of Florida, 1999.

[18] Peter F. Brown, Peter V. Desouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479, 1992.

[19] Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. Concreteness ratings for 40 thousand generally known English word lemmas. *Behavior research methods*, 46(3):904–911, 2014.

[20] Rafael A. Calvo and Sunghwan Mac Kim. Emotions in text: dimensional and categorical models. *Computational Intelligence*, 29(3):527–543, 2013.

[21] Erik Cambria, Amir Hussain, and Chris Eckl. Taking refuge in your personal sentic corner. In *Preceedings of The 5th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 35–43, 2011.

[22] Erik Cambria, Andrew Livingstone, and Amir Hussain. The hourglass of emotions. In *Cognitive behavioural systems*, pages 144–157. Springer, 2012.

[23] Erik Cambria, Daniel Olsher, and Dheeraj Rajagopal. SenticNet 3: a common and common-sense knowledge base for cognition-driven sentiment analysis. 2014.

[24] Erik Cambria, Soujanya Poria, Rajiv Bajpai, and Björn Schuller. SenticNet 4: A semantic resource for sentiment analysis based on conceptual primitives. In *Preceedings of the 26th International Conference on Computational Linguistics (COLING)*, pages 2666–2677, Osaka, Japan, 2016.

[25] Erik Cambria, Robert Speer, Catherine Havasi, and Amir Hussain. SenticNet: A Publicly Available Semantic Resource for Opinion Mining. volume 10, page 02, 2010.

[26] Lea Canales and Patricio Martínez-Barco. Emotion Detection from text: A Survey. *Proceedings of the Workshop on Natural Language Processing in the 5th Information Systems Research Working Days (JISIC)*, pages 37–43, 2014. 00000.

[27] Theodora Chaspari, Dimitrios Dimitriadis, and Petros Maragos. Emotion classification of speech using modulation features. pages 1552–1556. IEEE, 2014.

[28] François-Régis Chaumartin. UPAR7: A knowledge-based system for headline sentiment tagging. pages 422–425. Association for Computational Linguistics, 2007.

[29] Jifan Chen, Kan Chen, Xipeng Qiu, Qi Zhang, Xuanjing Huang, and Zheng Zhang. Learning Word Embeddings from Intrinsic and Extrinsic Views. *CoRR*, abs/1608.05852, 2016.

[30] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. pages 160–167. ACM, 2008.

[31] Bart Desmet and Véronique Hoste. Emotion detection in suicide notes. *Expert Systems with Applications*, 40(16):6351–6358, 2013.

[32] Paul Ekman. Facial expression and emotion. *American psychologist*, 48(4):384, 1993.

[33] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[34] Meghdad Farahmand, Aaron Smith, and Joakim Nivre. A Multiword Expression Data Set: Annotating Non-Compositionality and Conventionalization for English Noun Compounds. In *In Proceedings of the 11th Workshop on Multiword Expressions, NAACL*, pages 29–33, Denver, Colorado, 2015.

[35] Manaal Faruqui. *Diverse Context for Learning Word Representations*. Ph.D., University of Trento, 2016.

[36] Manaal Faruqui and Chris Dyer. Improving Vector Space Word Representations Using Multilingual Correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 462–471, Gothenburg, Sweden, 2014. Association for Computational Linguistics.

[37] J. R. Fontaine, K. R. Scherer, E. B. Roesch, and P. C. Ellsworth. The world of emotions is not two-dimensional. *Psychol Sci*, 18(12):1050–7, December 2007.

[38] Gottlob Frege. *The Foundations of Arithmetic*, volume Trans. J. L. Austin. Northwestern University Press, Evanston, Illinois, 2nd edition, 1884.

[39] Nico H Frijda. *The emotions*. Cambridge University Press, 1986.

[40] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. PPDB: The Paraphrase Database. In *Proceedings of NAACL-HLT*, pages 758–764, Atlanta, Georgia, June 2013.

[41] L. Gatti, m. guerini, and M. Turchi. SentiWords: Deriving a High Precision and High Coverage Lexicon for Sentiment Analysis. *IEEE Transactions on Affective Computing*, 7(4):409–421, 2016.

[42] Donald Glowinski, Antonio Camurri, Gualtiero Volpe, Nele Dael, and Klaus Scherer. Technique for automatic emotion recognition by body gesture analysis. pages 1–6. IEEE, 2008.

[43] Hongyu Gong, Suma Bhat, and Pramod Viswanath. Geometry of Compositionality. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, California, USA, 2017.

[44] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[45] Jonathan Gratch and Stacy Marsella. A domain-independent framework for modeling emotion. *Cognitive Systems Research*, 5(4):269–306, 2004.

[46] Alex Graves. Generating Sequences With Recurrent Neural Networks. *arXiv:1308.0850 [cs]*, August 2013. 00069.

[47] William L. Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. Inducing Domain-Specific Sentiment Lexicons from Unlabeled Corpora. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 595–605, Austin, Texas, USA, 2016.

[48] Meng-Ju Han, Chia-How Lin, and Kai-Tai Song. Autonomous Emotional Expression Generation of a Robotic Face. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, San Antonio, TX, USA, 11-14 October 2009*, pages 2427–2432, 2009.

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

[50] Kazuma Hashimoto and Yoshimasa Tsuruoka. Adaptive Joint Learning of Compositional and Non-Compositional Phrase Embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 205–215, Berlin, Germany, 2016. Association for Computational Linguistics.

[51] David R. Heise. Semantic differential profiles for 1,000 most frequent English words. *Psychological Monographs: General and Applied*, 79(8):1, 1965.

[52] David R. Heise. Affect control theory: Concepts and model. *The Journal of Mathematical Sociology*, 13(1-2):1–33, 1987.

[53] David R. Heise. *Expressive Order: Confirming Sentiments in Social Actions*. Springer US, Boston, MA, 2007. 00187.

[54] David R. Heise. *Surveying Cultures: Discovering Shared Conceptions and Sentiments*. John Wiley & Sons, March 2010. 00081.

[55] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[56] Jesse Hoey, Tim Schroder, and Areej Alhothali. Bayesian affect control theory. In *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*, pages 166–172. IEEE, 2013. 00020.

[57] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, pages 168–177, 2004.

[58] Eric Huang, Richard Socher, Christopher Manning, and Andrew Ng. Improving Word Representations via Global Context and Multiple Word Prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 873–882, Jeju Island, Korea, 2012. Association for Computational Linguistics.

[59] C. J. Hutto and Eric Gilbert. VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the Eighth International Conference on Weblogs and Social Media (ICWSM)*, 2014.

[60] Stephanie L. Hyland, Theofanis Karaletsos, and Gunnar Rätsch. A Generative Model of Words and Relationships from Multiple Sources. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2622–2629, Phoenix, Arizona, USA., 2016.

[61] Suwicha Jirayucharoensak, Setha Pan-Ngum, and Pasin Israsena. EEG-Based Emotion Recognition Using Deep Learning Network with Principal Component Based Covariate Shift Adaptation. *The Scientific World Journal*, 2014, 2014.

[62] Daniel Jurafsky and James H. Martin. *Speech and Language Processing - An introduction to NLP, CL, SR*. Prentice Hall, May 2008.

[63] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-Thought Vectors. In *Proceedings of NIPS, 2015, Montreal, Quebec, Canada*, pages 3294–3302, 2015.

[64] Ioannis Korkontzelos. *Unsupervised learning of multiword expressions*. PhD thesis, University of York, UK, 2010.

[65] Ioannis Korkontzelos, Torsten Zesch, Fabio Massimo Zanzotto, and Chris Biemann. SemEval-2013 Task 5: Evaluating Phrasal Semantics. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, pages 39–47, Atlanta, Georgia, USA, 2013. Association for Computational Linguistics.

[66] George Lakoff. Linguistic gestalts. In *Papers from the... Regional Meeting. Chicago Ling. Soc. Chicago, Ill.*, volume 13, pages 236–287, 1977.

[67] Richard S Lazarus. *Emotion and adaptation*. Oxford University Press, 1991.

[68] Quoc V. Le and Tomas Mikolov. Distributed Representations of Sentences and Documents. In *Proceedings of ICML, Beijing, China, 21-26*, pages 1188–1196, 2014.

[69] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. 00000.

[70] Sophia Yat Mei Lee and Zhongqing Wang. Emotion in code-switching texts: Corpus construction and analysis. *ACL-IJCNLP 2015*, page 91, 2015.

[71] Geoffrey N. Leech. *Semantics: The Study of Meaning*. Penguin Books, 2 edition, 1981.

[72] Omer Levy and Yoav Goldberg. Dependency-Based Word Embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland, 2014. Association for Computational Linguistics.

[73] Omer Levy and Yoav Goldberg. Neural Word Embedding as Implicit Matrix Factorization. In *Advances in Neural Information Processing Systems*, pages 2177–2185, Montreal, Quebec, Canada, 2014.

[74] Omer Levy, Yoav Goldberg, and Ido Dagan. Improving Distributional Similarity with Lessons Learned from Word Embeddings. *Transactions of the Association of Computational Linguistics*, 3:211–225, 2015.

[75] Chengxin Li, Huimin Wu, and Qin Jin. Emotion Classification of Chinese Microblog Text via Fusion of BoW and eVector Feature Representations. In *Natural Language Processing and Chinese Computing*, pages 217–228. Springer, 2014.

[76] Minglei Li, Yunfei Long, and Qin Lu. A regression approach to valence-arousal ratings of words from word embedding. In *Proceedings of the 2016 International Conference on Asian Language Processing (IALP)*, pages 120–123, Tainan, Taiwan, 2016. IEEE.

[77] Minglei Li, Yunfei Long, Qin Lu, and Wenjie Li. Emotion Corpus Construction Based on Selection from Hashtags. In *Proceedings of International Conference on Language Resources and Evaluation (LREC)*, Portorož, Slovenia, 2016.

[78] Minglei Li, Qin Lu, and Yunfei Long. Are Manually Prepared Affective Lexicons Really Useful for Sentiment Analysis. *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2:146–150, 2017.

[79] Minglei LI, Qin Lu, and Yunfei Long. Representation Learning of Multiword Expressions with Compositionality Constraint. In *Proceedings of the 10th International Conference on Knowledge Science, Engineering and Management (KSEM)*, Melbourne, Australia, 2017.

[80] Minglei Li, Qin Lu, Yunfei Long, and Lin Gui. Affective State Prediction of Contextualized Concepts. In Neil Lawrence and Mark Reid, editors, *Proceedings of IJCAI 2017 Workshop on Artificial Intelligence in Affective Computing*, volume 66 of *Proceedings of Machine Learning Research*, pages 45–57. PMLR, August 2017.

[81] Minglei Li, Qin Lu, Yunfei Long, and Lin Gui. Inferring Affective Meanings of Words from Word Embedding. *IEEE Transactions on Affective Computing*, 8(4):443–456, October 2017.

[82] Shaohua Li, Jun Zhu, and Chunyan Miao. A Generative Word Embedding Model and its Low Rank Positive Semidefinite Solution. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1599–1609, Lisbon, Portugal, 2015. Association for Computational Linguistics.

[83] Percy Liang. *Semi-supervised learning for natural language*. PhD thesis, Massachusetts Institute of Technology, 2005.

[84] Bing Liu. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167, 2012.

[85] Bing Liu. *Sentiment analysis: mining opinions, sentiments, and emotions*. Cambridge University Press, New York, NY, 2015.

[86] Hugo Liu, Henry Lieberman, and Ted Selker. A model of textual affect sensing using real-world knowledge. pages 125–132. ACM, 2003.

[87] Dermot Lynott and Louise Connell. Modality exclusivity norms for 423 object properties. *Behavior Research Methods*, 41(2):558–564, 2009.

[88] Dermot Lynott and Louise Connell. Modality exclusivity norms for 400 nouns: The relationship between perceptual experience and surface word form. *Behavior research methods*, 45(2):516–526, 2013.

[89] Hugo Lövheim. A new three-dimensional model for emotions and monoamine neurotransmitters. *Medical Hypotheses*, 78(2):341–348, 2012.

[90] Albert Mehrabian. Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament. *Current Psychology*, 14(4):261–292, 1996.

[91] Oren Melamud, David McClosky, Siddharth Patwardhan, and Mohit Bansal. The Role of Context Types and Dimensionality in Learning Word Embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1030–1040, San Diego, California, 2016. Association for Computational Linguistics.

[92] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3781, 2013.

[93] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Proceedings of 11th Annual Conference of the International Speech Communication Association*, pages 1045–1048, Makuhari, Chiba, Japan, 2010.

[94] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of Advances in Neural Information Processing Systems*, pages 3111–3119, Lake Tahoe, Nevada, United States, 2013.

[95] Lynn C. Miller, Stephen J. Read, Wayne Zachary, and Andrew Rosoff. Modeling the impact of motivation, personality, and emotion on social behavior. In *International Conference on Social Computing, Behavioral Modeling, and Prediction*, pages 298–305. Springer, 2010.

[96] Jeff Mitchell and Mirella Lapata. Vector-based Models of Semantic Composition. In *Proceedings of ACL-08: HLT*, pages 236–244, Columbus, Ohio, 2008. Association for Computational Linguistics.

[97] Jeff Mitchell and Mirella Lapata. Composition in Distributional Models of Semantics. *Cognitive Science*, 34(8):1388–1429, November 2010.

[98] Andriy Mnih and Geoffrey Hinton. Three new graphical models for statistical language modelling. In *Proceedings of the 24th international conference on Machine learning*, pages 641–648. ACM, 2007.

[99] Saif Mohammad. Portable features for classifying emotional text. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 587–591. Association for Computational Linguistics, 2012.

[100] Saif M Mohammad. # Emotional tweets. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 246–255. Association for Computational Linguistics, 2012.

[101] Saif M Mohammad. Sentiment Analysis of Social Media Texts. Technical report, EMNLP, 2014. 00000.

[102] Saif M. Mohammad. Sentiment Analysis: Detecting Valence, Emotions, and Other Affectual States from Text. In Herb Meiselman, editor, *Emotion Measurement*. Elsevier, 2016.

[103] Saif M Mohammad and Svetlana Kiritchenko. Using hashtags to capture fine emotion categories from tweets. *Computational Intelligence*, 31(2):301–326, 2015.

[104] Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the 7th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT*, pages 321–327, 2013.

[105] Saif M Mohammad and Peter D Turney. Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon. pages 26–34. Association for Computational Linguistics, 2010.

[106] Saif M Mohammad and Peter D Turney. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465, 2013.

[107] Antonio Moreno-Ortiz, Chantal Pérez-Hernández, and M. Ángeles Del-Olmo. Managing Multiword Expressions in a Lexicon-Based Sentiment Analysis System for Spanish. In *Proceedings of the 9th Workshop on Multiword Expressions, MWE@NAACL-HLT*, volume 13, pages 1–10, Atlanta, Georgia, USA, 2013.

[108] Preslav Nakov, Zornitsa Kozareva, Alan Ritter, Sara Rosenthal, Veselin Stoyanov, and Theresa Wilson. Semeval-2013 task 2: Sentiment analysis in twitter. In *Proceedings of the 7th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2013*, pages 312–320, Atlanta, Georgia, USA, 2013.

[109] Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. Affect analysis model: novel rule-based approach to affect sensing from text. *Natural Language Engineering*, 17(01):95–135, 2011.

[110] Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. Integrating Distributional Lexical Contrast into Word Embeddings for Antonym-Synonym Distinction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 454–459, Berlin, Germany, 2016. Association for Computational Linguistics.

[111] Andrew Ortony. *The cognitive structure of emotions*. Cambridge university press, 1990.

[112] Charles Egerton Osgood, George J. Suci, and Percy H. Tannenbaum. *The Measurement of Meaning*. University of Illinois Press, 1957.

[113] Bo Pang and Lillian Lee. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 115–124, University of Michigan, USA, 2005.

[114] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

[115] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. pages 79–86. Association for Computational Linguistics, 2002.

[116] W Gerrod Parrott. *Emotions in social psychology: Essential readings*. Psychology Press, 2001.

[117] Barbara Partee. Lexical semantics and compositionality. *An invitation to cognitive science: Language*, 1:311–360, 1995.

[118] Ted Pedersen, Siddharth Patwardhan, Jason Michelizzi, and Satanjeev Banerjee. *Wordnet:: similarity*. 2005. 00026.

[119] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, 2014. Association for Computational Linguistics.

[120] R. W. Picard. Affective Computing. Technical Report 321, MIT Media Lab, 20 Ames St., Cambridge, MA 02139, 1995.

[121] Changqin Quan and Fuji Ren. A blog emotion corpus for emotional expression analysis in Chinese. *Computer Speech & Language*, 24(4):726–749, 2010.

[122] Pushpendre Rastogi, Benjamin Van Durme, and Raman Arora. Multiview LSA: Representation Learning via Generalized CCA. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 556–566, Denver, Colorado, 2015. Association for Computational Linguistics.

[123] Niklas Ravaja, Timo Saari, Marko Turpeinen, Jari Laarni, Mikko Salminen, and Matias Kivikangas. Spatial presence and emotions during video game playing: Does it matter with whom you play? *Presence: Teleoperators and Virtual Environments*, 15(4):381–392, 2006.

[124] Siva Reddy, Diana McCarthy, and Suresh Manandhar. An Empirical Study on Compositionality in Compound Nouns. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 210–218, Chiang Mai, Thailand, 2011. Asian Federation of Natural Language Processing.

[125] Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif M. Mohammad, Alan Ritter, and Veselin Stoyanov. Semeval-2015 task 10: Sentiment analysis in twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015*, pages 451–463, 2015.

[126] Sascha Rothe, Sebastian Ebert, and Hinrich Schütze. Ultradense Word Embeddings by Orthogonal Transformation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 767–777, San Diego California, USA, 2016. The Association for Computational Linguistics.

[127] James A Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980.

[128] Bahar Salehi, Paul Cook, and Timothy Baldwin. A Word Embedding Approach to Predicting the Compositionality of Multiword Expressions. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 977–983, Denver, Colorado, USA, 2015. Association for Computational Linguistics.

[129] Inaki San Vicente, Rodrigo Agerri, German Rigau, and Donostia-San Sebastián. Simple, Robust and (almost) Unsupervised Generation of Polarity Lexicons for Multiple Languages. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 88–97, Gothenburg, Sweden, 2014. The Association for Computer Linguistics.

[130] K. R. Scherer and H. G. Wallbott. Evidence for universality and cultural variation of differential emotion response patterning. *Journal of Personality and Social Psychology*, 66(2):310–328, February 1994.

[131] Klaus R. Scherer. Psychological models of emotion. *The neuropsychology of emotion*, 137(3):137–162, 2000.

[132] David S. Schmidtke, Tobias Schröder, Arthur M. Jacobs, and Markus Conrad. ANGST: Affective norms for German sentiment terms, derived from the affective norms for English words. *Behavior Research Methods*, 46(4):1108–1118, January 2014.

[133] Nathan Schneider, Spencer Onuffer, Nora Kazour, Emily Danchik, Michael T. Mordowanec, Henrietta Conrad, and Noah A. Smith. Comprehensive Annotation of Multiword Expressions in a Social Web Corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 455–461, Reykjavik, Iceland, 2014.

[134] Marc Schröder. Emotional speech synthesis: a review. In *EUROSPEECH 2001 Scandinavia, 7th European Conference on Speech Communication and Technology, 2nd INTERSPEECH Event, Aalborg, Denmark, September 3-7, 2001*, pages 561–564, 2001.

[135] H. Schütze. Dimensions of Meaning. In *Proceedings of the 1992 ACM/IEEE Conference on Supercomputing*, Supercomputing '92, pages 787–796, Los Alamitos, CA, USA, 1992. IEEE Computer Society Press.

[136] Vered Shwartz, Yoav Goldberg, and Ido Dagan. Improving Hypernymy Detection with an Integrated Path-based and Distributional Method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2389–2398, Berlin, Germany, 2016.

[137] Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the*

2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 1201–1211, Jeju Island, Korea, 2012. Association for Computational Linguistics.

[138] Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th International Conference on Machine Learning*, pages 129–136, Bellevue, Washington, USA, 2011.

[139] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, 2013.

[140] Kaisong Song, Shi Feng, Wei Gao, Daling Wang, Ling Chen, and Chengqi Zhang. Build Emotion Lexicon from Microblogs by Combining Effects of Seed Words and Emoticons in a Heterogeneous Graph. In *Proceedings of the 26th ACM Conference on Hypertext & Social Media*, pages 283–292. ACM, 2015.

[141] Robert Speer and Joshua Chin. An Ensemble Method to Produce High-Quality Word Embeddings. *CoRR*, abs/1604.01692, 2016.

[142] Robert Speer, Joshua Chin, and Catherine Havasi. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 4444–4451, San Francisco, California, USA., 2017.

[143] Robert Speer and Catherine Havasi. Representing General Relational Knowledge in ConceptNet 5. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC)*, pages 3679–3686, Istanbul, Turkey, 2012.

[144] Jacopo Staiano and Marco Guerini. Depeche Mood: a Lexicon for Emotion Analysis from Crowd Annotated News. In *Proc. Ann. Meeting of the Assoc. for Computational Linguistics (ACL)*, volume 2, pages 427–433, Baltimore, MD, USA, 2014.

[145] Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilvie. The General Inquirer: A Computer Approach to Content Analysis. *Journal of Regional Science*, 8(1):113–116, 1968.

[146] Carlo Strapparava and Rada Mihalcea. Semeval-2007 task 14: Affective text. pages 70–74. Association for Computational Linguistics, 2007.

[147] Fei Sun, Jiafeng Guo, Yanyan Lan, Jun Xu, and Xueqi Cheng. Inside Out: Two Jointly Predictive Models for Word Representations and Phrase Representations. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2821–2827, Phoenix, Arizona, USA, 2016.

[148] Xiao Sun, Chengcheng Li, and Jiaqi Ye. Chinese microblogging emotion classification based on support vector machine. In *Computing, Communication and Networking Technologies (ICCCNT), 2014 International Conference on*, pages 1–5. IEEE, 2014.

[149] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112, 2014.

[150] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[151] Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432, 2015.

[152] Duyu Tang, Bing Qin, Ting Liu, and Zhenghua Li. Learning Sentence Representation for Emotion Classification on Microblogs. In *Natural Language Processing and Chinese Computing*, pages 212–223. Springer, 2013.

[153] Duyu Tang, Bing Qin, Ting Liu, and Qiuhui Shi. Emotion Analysis Platform on Chinese Microblog. *arXiv preprint arXiv:1403.7335*, 2014.

[154] Duyu Tang, Furu Wei, Bing Qin, Ming Zhou, and Ting Liu. Building Large-Scale Twitter-Specific Sentiment Lexicon : A Representation Learning Approach. In *Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers (COLING)*, pages 172–182, Dublin, Ireland, 2014.

[155] Jianhua Tao and Tieniu Tan. Affective computing: A review. In *International Conference on Affective computing and intelligent interaction*, pages 981–995. Springer, 2005.

[156] Duy Tin Vo and Yue Zhang. Don't Count, Predict! An Automatic Approach to Learning Sentiment Lexicons for Short Text. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 2, Berlin, Germany, 2016.

[157] Silvan S Tomkins. Affect theory. *Approaches to emotion*, 163:195, 1984.

[158] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics, 2010.

[159] Peter D. Turney. Domain and Function: A Dual-Space Model of Semantic Relations and Compositions. *CoRR*, abs/1309.4035, 2013.

[160] Peter D. Turney and Michael L. Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)*, 21(4):315–346, 2003.

[161] Peter D. Turney and Patrick Pantel. From Frequency to Meaning: Vector Space Models of Semantics. *J. Artif. Intell. Res.*, 37:141–188, 2010.

[162] Leonid Velikovich, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan McDonald. The viability of web-derived polarity lexicons. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 777–785. Association for Computational Linguistics, 2010.

[163] Shan Wang and Francis Bond. Building the chinese open wordnet (cow): Starting from core synsets. pages 10–18. Citeseer, 2013.

[164] Sida Wang and Christopher D Manning. Baselines and bigrams: Simple, good sentiment and topic classification. pages 90–94. Association for Computational Linguistics, 2012.

[165] Wenbo Wang, Lu Chen, Krishnaprasad Thirunarayan, and Amit P. Sheth. Harnessing twitter" big data" for automatic emotion identification. In *Privacy, Security, Risk and Trust (PASSAT),2012 International Confernece on Social Computing (SocialCom)*, pages 587–592. IEEE, 2012.

[166] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge Graph and Text Jointly Embedding. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1591–1601, Doha, Qatar, 2014. Association for Computational Linguistics.

[167] Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. Norms of valence, arousal, and dominance for 13,915 English lemmas. *Behavior research methods*, 45(4):1191–1207, 2013.

[168] Wen-Li Wei, Chung-Hsien Wu, and Jen-Chun Lin. A regression approach to affective rating of chinese words from ANEW. In *Affective Computing and Intelligent Interaction*, pages 121–131. Springer Berlin Heidelberg, 2011.

[169] Cynthia Whissell. The dictionary of affect in language. *Emotion: Theory, research, and experience*, 4(113-131):94, 1989.

[170] Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating Expressions of Opinions and Emotions in Language. *Language Resources and Evaluation*, 39(2-3):165–210, 2005.

[171] John Wieting, Mohit Bansal, Kevin Gimpel, Karen Livescu, and Dan Roth. From Paraphrase Database to Compositional Paraphrase Model and Back. *Transactions of the Association for Computational Linguistics (TACL)*, 3:345–358, 2015.

[172] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 347–354, 2005.

[173] Chung-Hsien Wu, Ze-Jing Chuang, and Yu-Chung Lin. Emotion recognition from text using semantic labels and separable mixture models. *ACM transactions on Asian language information processing (TALIP)*, 5(2):165–183, 2006.

[174] Jun Xu, Ruifeng Xu, Qin Lu, and Xiaolong Wang. Coarse-to-fine sentence-level emotion classification based on the intra-sentence features and sentential context. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2455–2458. ACM, 2012.

[175] Linhong Xu, Hongfei Lin, Pan Yu, Hui Ren, and Jianmei Chen. Constructing the Affective Lexicon Ontology. *Journal of the China Society for Scientific and Technical Information*, 2:006, 2008.

[176] Jasy Liew Suet Yan, Howard R. Turtle, and Elizabeth D. Liddy. EmoTweet-28: a fine-grained emotion corpus for sentiment analysis. 2016.

[177] Changhua Yang, Kevin Hsin-Yih Lin, and Hsin-Hsi Chen. Emotion Classification Using Web Blog Corpora. pages 275–278, 2007.

[178] Min Yang, Baolin Peng, Zheng Chen, Dingju Zhu, and Kam-Pui Chow. A Topic Model for Building Fine-grained Domain-specific Emotion Lexicon. In *ACL*, 2014.

[179] Yuanlin Yao, Ruifeng Xu, Qin Lu, Bin Liu, Jun Xu, Chengtian Zou, Li Yuan, Shuwei Wang, Lin Yao, and Zhenyu He. Reader emotion prediction using concept and concept sequence features in news headlines. In *Computational Linguistics and Intelligent Text Processing*, pages 73–84. Springer, 2014.

[180] Majid Yazdani, Meghdad Farahmand, and James Henderson. Learning Semantic Composition to Detect Non-compositionality of Multiword Expressions. In *Proceedings of EMNLP*, pages 1733–1742, 2015.

[181] Wenpeng Yin and Hinrich Schütze. An Exploration of Embeddings for Generalized Phrases. In *Proceedings of the ACL, Student Research Workshop*, pages 41–47, 2014.

[182] Wenpeng Yin and Hinrich Schütze. Learning Word Meta-Embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*

(Volume 1: Long Papers), pages 1351–1360, Berlin, Germany, 2016. Association for Computational Linguistics.

[183] Liang-Chih Yu, Lung-Hao Lee, Shuai Hao, Jin Wang, Yunchao He, Jun Hu, K. Robert Lai, and Xuejie Zhang. Building Chinese Affective Resources in Valence-Arousal Dimensions. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 540–545, 2016.

[184] Liang-Chih Yu, Jin Wang, K. Robert Lai, and Xue-jie Zhang. Predicting Valence-Arousal Ratings of Words Using a Weighted Graph Method. In *Proc. Ann. Meeting of the Assoc. for Computational Linguistics (ACL)*, volume 2, pages 788–793, 2015.

[185] Mo Yu and Mark Dredze. Learning Composition Models for Phrase Embeddings. *Transactions of the Association for Computational Linguistics*, 3:227–242, 2015.

[186] YAO Yuanlin, WANG Shuwei, XU Ruifeng, LIU Bin, GUI Lin, LU Qin, and WANG Xiaolong. The Construction of an Emotion Annotated Corpus on Microblog Text. *Journal of Chinese Information Processing*, 28(5):83–91, 2014.

[187] Jingwei Zhang, Jeremy Salwen, Michael Glass, and Alfio Gliozzo. Word Semantic Representations using Bayesian Probabilistic Tensor Factorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1522–1531, Doha, Qatar, 2014. Association for Computational Linguistics.

[188] Sicheng Zhao, Yue Gao, Xiaolei Jiang, Hongxun Yao, Tat-Seng Chua, and Xiaoshuai Sun. Exploring principles-of-art features for image emotion recognition. pages 47–56. ACM, 2014.

[189] Yu Zhao, Zhiyuan Liu, and Maosong Sun. Phrase Type Sensitive Tensor Indexing Model for Semantic Composition. In *Proceedings of the Twenty-Ninth {AAAI} Conference on Artificial Intelligence*, pages 2195–2202, Austin, Texas, USA, 2015.

[190] Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. Emotional Chatting Machine: Emotional Conversation Generation with Internal and External Memory. In *Proceedings of EMNLP*, 2017. arXiv: 1704.01074.

[191] Xiaodan Zhu, Svetlana Kiritchenko, and Saif M. Mohammad. NRC-Canada-2014: Recent Improvements in the Sentiment Analysis of Tweets. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING*, pages 443–447, Dublin, Ireland, 2014.