A STUDY ON USING PERSONAL PROFILES FOR A BIASED READER EMOTION PREDICTION MODEL

YUNFEI LONG

PhD

The Hong Kong Polytechnic University

2019

The Hong Kong Polytechnic University
Department of Computing

A Study on Using Personal Profiles for a Biased Reader Emotion Prediction Model

YUNFEI LONG

A thesis submitted in partial fulfilment of the requirements

for the degree of Doctor of Philosophy

June 2018

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)

Yunfei Long (Name of student)

Abstract

In the age of social media, users express their personal feelings and emotions through the Web. In addition to understanding the emotion of the public, it is also important to learn how individual subjectivity and their bias affect emotion analysis especially in social media and review texts. The main objective of this study is to investigate the effect of personal profiles in emotion analysis.

This thesis focuses on emotion analysis from social media and review text, and studies four areas in subjectivity linked emotion analysis, including (1) improving emotion analysis from cognitive perspective by identifying linguistic features more appropriate for social media text, (2) using cognition grounded data to improve emotion prediction models, (3) learning the representation of user profiles by addressing the data sparseness through two methods, and (4) incorporating user profiles into emotion analysis model to take subjectivity as a bias into consideration.

Based on the premise that emotion is a personalized cognitive process encoded in different types of linguistic features, we first explore additional linguistic features relevant to social media and review text. In addition to the traditional lexical features, we propose a linguistic-driven model to explore the use of morpho-syntactic features such as passive construction, verb order as well as some less explored orthographic text features such as unusual use of punctuations and code-switches which are often seen in this genre of text. Evaluation on both a personally generated micro-blog dataset and a formal news collection shows that incorporating our proposed linguistic features can improve emotion analysis by introducing genre and stylist information encoded in social media text.

Cognitive studies support the linguistic fact that not all words contribute equally to the semantic and affective meaning of sentences. Some words are more important than others in conveying semantic meanings. Computational attention models are proposed to give different weights to different words in text. However, many attention models are

built using local text features through distributional similarity which lack the theoretical foundation to reflect the cognitive basis. This motivates us to explore the use of eye tracking data as cognition based information to train attention models to further improve the performance of linguistic-driven models. Our proposed method can capture attention of words more comprehensively using a two level approach. Evaluations show that our method outperforms the state-of-the-art methods significantly. We prove that cognition grounded data can be used to improve attention mechanisms and thus indirectly improves the performance of sentiment analysis.

Presenting user profile using dense vector representation through user activities is the key to build user profiling models. However, like many social media data, user activities follow the long-tail distribution. Thus, the key to obtain a better representation of user profiles is to address the data sparseness issue. Inspired by the stimulus generalization theory and the halo effect in cognitive science, we first propose a novel approach to predict user preferences by learning from both observed comments and missing comments based on the missing-not-at-random hypothesis. Then we explore methods to extend context for user profiles through network links in social media data. We propose a novel approach to learn node embedding through a joint learning framework of both network links and text associated with nodes. The method can handle both homogeneous networks and heterogeneous networks with multiple types of links. A novel attention mechanism is also proposed to make good use of text extended through links to obtain a larger network context.

Finally, emotion as a cognitive process is largely subjective and user bias plays a significant role in emotion analysis. Lenient users tend to give higher ratings than finicky ones even if they review the same products with similar wording, On the other hand, popular products do receive higher ratings than those unpopular ones because the aggregation of user reviews still shows the difference in opinions for different products. In this work, we propose a deep learning method to incorporate biased user profile into emotion analysis for review text. Individual user bias as user profiles, is learned through a neural network model. Then, user profiles as a collection are aggregated by a memory network to encode the user bias. A separate memory network is also used to learn product information. In this way, user profiles and product information can be captured more efficiently as they

are different by nature. Lastly, the dual memory networks is merged into a unified classification model for joint optimization. Evaluations on three commonly used benchmark datasets show that our dual memory network model is more effective than the state-of-art methods.

List of Publications

Paper already published

As first author or co-first author:

- **Yunfei Long**, Qin Lu, Yue Xiao, MingLei Li, and Chu-Ren Huang: Domain-specific user preference prediction based on multiple user activities. In IEEE International Conference on Big Data (IEEE Bigdata 2016), pp. 3913-3921. IEEE, 2016.

- **Yunfei Long**, Dan Xiong, Qin Lu, Minglei Li, and Chu-Ren Huang: Named Entity Recognition for Chinese Novels in the Ming-Qing Dynasties. In Workshop on Chinese Lexical Semantics (CLSW 2016), pp. 362-375. Springer, Cham, 2016.

- **Yunfei Long**, Qin Lu, Rong Xiang, Minglei Li, and Chu-Ren Huang: A Cognition Based Attention Model for Sentiment Analysis. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017), pp. 462-471. 2017.

- **Yunfei Long**, Qin Lu, Rong Xiang, Minglei Li, and Chu-Ren Huang: Fake News Detection Through Multi-Perspective Speaker Profiles. In Proceedings of the Eighth International Joint Conference on Natural Language Processing (IJCNLP 2017)(Volume 2: Short Papers), vol. 2, pp. 252-256. 2017.

- Chen, I-Hsuan, **Yunfei Long**, Qin Lu, and Chu-Ren Huang: Leveraging Eventive Information for Better Metaphor Detection and Classification. In Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), pp. 36-46. 2017.

- **Yunfei Long**, Mingyu Ma, Qin Lu, Rong Xiang, and Chu-Ren Huang: Dual Memory Network Model for Biased Product Review Classification. arXiv preprint arXiv:1809.05807 (Accepted by 2018 EMNLP 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis).

- **Yunfei Long**, Rong Xiang, Qin Lu, Xiong Dan, Chenlin Bi, and Chu-Ren Huang: Learning Heterogeneous Network Embedding from Text and Links, IEEE Access. Volume: 6, pp 55850-55860. 2018.

- I-Hsuan Chen, **Yunfei Long**, Chu-Ren Huang, Qin Lu and Rong Xiang: Leveraging Orthographic features for Sentiment Classification in Texts varying in Sentence Length, Accepted by IEEE Access, 2019.

- I-Hsuan Chen, Qingqing Zhao, **Yunfei Long**, Chu-Ren Huang, and Qin Lu: Mandarin Chinese Modality Exclusivity Norms, Accepted by Plus One, 2019.

As corresponding author:

- Minglei Li, Qin Lu, Dan Xiong, and **Yunfei Long**: Phrase Embedding Learning Based on External and Internal Context with Compositionality Constraint. *Knowledge-Based Systems 152 (2018)*: 107-116.

As a minor contributor:

- 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 2016)*, 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 2016)*, Portorož, Slovenia, 2016.

- Minglei Li, Qin Lu, **Yunfei Long** and Lin Gui: Inferring Affective Meanings of Words from Word Embedding. *Journal of IEEE Transactions on Affective Computing*(TAC), 2017.

- 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*(IJCNLP 2017)(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*(IJCAI AffComp 2017), Melbourne, Australia, 2017.

- Minglei Li, Qin Lu, **Yunfei Long**: Representation Learning of Multi-word 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.

- Zhao Qingqing, Chu-Ren Huang, and **Yunfei Long**: Synaesthesia in Chinese: A corpus-based study on gustatory adjectives in Mandarin. *Linguistics*, 56(5), 1167-1194.

- Rong Xiang, **Yunfei Long**, Qin Lu, Dan Xiong, and I-Hsuan Chen: Leveraging Writing Systems Change for Deep Learning Based Chinese Emotion Analysis. In

Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp. 91-96. 2018.

List of Papers under review

- **Yunfei Long**, Qin Lu, Minglei Li, Rong Xiang, and Chu-Ren Huang: Improving attention model based on cognition grounded data for sentiment analysis, Submitted to IEEE Transactions on Affective Computing (Nov, 2017, under minor revision)

- Minglei Li, Qin Lu, Dan Xiong, and **Yunfei Long**: Event Role Level Emotion Analysis Based on LSTM with Word Embedding, Submitted to IEEE Transaction on Affective Computing (April 2018)

Acknowledgements

Stay hungry, stay foolish. The most important thing I learned during my PhD study is about how to obtain knowledge at the frontier of human wisdom. The journey is hard, but Difficulty is the nurse of greatness.

I would give my deepest gratitude to my supervisor, Professor Qin Lu for her guidance, great support and kind advice throughout my PhD studies. At many stages in the course of this research project, I benefited from her advice, particularly when exploring new ideas. Her positive outlook and confidence in me inspired me to explore with no hesitation. I would also like to thank her for her tireless help in structuring my works and polish my thesis. My English improved a lot as a result of working under her supervision.

I would like to thank my co-supervisor, Prof Chu-Ren Huang for his constant support, availability, and constructive suggestion, which were determinant for the accomplishment of the work presented in this thesis.

I would also like to thank my Board of Examiner members, Prof Grace Ngai, Prof Helen Meng, and Prof Timothy Baldwin, for their valuable comments to polish my thesis.

A project of this nature, based on both emotion analysis and user modeling, is only possible with the help of many people. They too have played their part in the development of the system. As I begin the journey to change from a Chinese literature student to an NLP researcher, I am much in debt to Mr Rong Xiang and Mr Yue Xiao for helping me with my engineering skills.

Special thanks to Dr I-Hsuan Chen and Dr Qingqing Zhao, who inspired me to build a linguistic-driven model for emotion analysis which resulted in the work described in Chapter 3.

I would also like to thank Dr Minglei Li and Dr Lin Gui for their insightful ideas about how to learn user profiles from sparse data which helped me to complete my work

in Chapter 4 and Chapter 5. Thanks to Mingyu Ma for helping me to prepare for the experiments in Chapter 6. Thanks to Wenhao Ying, Yufei Zheng, and Ying Jiao for their assistance and valuable discussions. Thanks to Dan Xiong for her friendship and kind sharing of daily life which made my life in Hong Kong much easier.

Thanks to all the staff in Department of Computing, specially those in the general office for being so kind and helpful. Thanks to all the friends I met in the last three years. The moment of leisure shared together with you all helped me to overcome some difficult moments in my study.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Text has been one of the most important social media content on the web for people to express information, exchange ideas, explain scientific discoveries and create stories, etc [96]. The growing popularity of social media has fundamentally changed the web from a simple information dissemination platform to an interactive social network based platform for information exchange and sharing as well as for personal expressions of individual feelings. Social media also serve as a platform for on-line emotional support. Emotions expressed in text through the Web, especially in different social media and review platforms, can affect its readers in such an unprecedented speed and scale which sometimes can have significant consequences. Generally speaking, text written in social media or for review purpose has two characteristics. First, they are rich in both emotion and subjectivity as they often reflect personal bias, attitude, and preference in their reaction to daily events or certain products. Secondly, there is also social network information which we can leverage. In this work, we focus on emotion analysis for this genre of text, which we refer to as **Social&Review (SR text)**. The characteristics of SR text make it possible to take into account of individual bias and social connections in emotion analysis.

The term **emotion** has many different definitions. In psychology study, Scherer et al.[196] give a formal definition of **emotion** as *episodes of coordinated changes in several components (including at least neurophysiology activation, motor expression, and subjec-*

tive feeling but possibly also action tendencies and cognitive processes) in response to external or internal events of major significance to the organism. Emotion can be represented by **discrete emotion models** such as *anger, disgust, fear, joy, sadness, surprise* [56], and **polarity** of *positive, negative* [103] or by **dimensional emotion models** using continuous values in every dimension such as the three-dimensional **Valence-Arousal-Dominance (VAD** model [192]. These **emotion representation models** are the basis for emotion analysis using computational means. **Emotion analysis** (**EA**) refers to computational methods to enable machines to predict or generate emotions like a human manner.

The terms **affect** is used to describe a collection of feelings and other traits of human beings including emotion, mood, personal stance, attitude, and personality traits [181, 196]. From this definition, it is easy to see that emotion is a subordinate concept of affect [121]. In natural language processing (NLP) study, **affective computing** refers to the study of computational methods to assign computers with human-like capabilities regarding observation, interpretation, and generation of affect features [220]. Emotion generation, emotion detection through facial expression and body gestures, emotion recognition from speech data etc. are all parts of affective computing [182]. This thesis focuses on emotion analysis of text, which is only a part of affective computing [11].

The term **emotion** and **sentiment** are closely related. Both emotions and sentiments refer to "experiences that result from the combined influences of the biological, the cognitive, and the social" [206]. In general, sentiment refers to one's attitude towards a particular target or topic [162]. The term **sentiment analysis** is most commonly used to refer to the task of automatically determining the polarity of a piece of text as a binary classification task. Sometimes, it can also be positive, negative, or neutral. Following the general definition of many industry experts, we regard emotional analysis as a more comprehensive, and evolved form of sentiment analysis [1]. In NLP community, approaches to build sentiment analysis models and emotion analysis models are quite similar. However, there

[1] https://www.linkedin.com/pulse/sentiment-analysis-versus-emotional-same-different-shahbaz-anwar/

are much less work in emotion analysis because emotion representation models are more complex than sentiment which is polarity based. This thesis works on both sentiment analysis and emotion analysis, and the two terms are thus used interchangeably when there is no ambiguity.

Emotion analysis has been widely applied in research and industry. Emotion analysis has many downstream applications, such as in analysis of consumer's response to product, service, advertisement to help for future decisions [21, 174], recommendation for entertainments such as movies, books, music or pictures that are suitable for users' current mood or desire [26], analysis of social responses to emergency events, such as a disaster, a war, a political event, news, etc [10], prediction of suicide tendency through social network [48], generation of appropriate responses with emotional recognition in dialog systems [186, 256], and emotion analysis for an assistant system [132].

Most emotion analysis methods treat training data as a collection of text [176]. Prediction is conducted on label of the text, which can be a document, a sentence, or an aspect of a product [199]. However, emotion as a cognitive process is largely subjective and user bias plays a significant role in emotion analysis. Lenient users tend to give higher ratings than finicky ones even if they review the same products with similar wording. This means using personal profile information has great potential to improve emotion analysis models for SR text. Recently, a few works start to incorporate global user profile and product characteristics into emotion analysis [214, 67, 33], but they treat different profile information in a unified model, therefore the difference in profile information is neglected.

In this thesis, the term **personal profile** is often referred to as **user profile** because the word "person" refers to individual users. So, we use the term **user profile** in the remaining part of this thesis. A user profile generally refers to a collection of information associated with a user including both social demographic information and personal preference information. **User background information** includes factual information such as gender, age, location, education, and marriage status etc. and **user preference** reflects the subjective

choices of a person such as his attitude and views towards the outside world with respect to entities, events, or social issues. A user profile is a general term, and specific information depends on the applications where such information can be obtained. Regardless of the content, the accuracy of a user profile depends on how user information is gathered and organized, and how accurately the information represent the user profile [41]. The process to gather, organize and interpret information for summarizing and describing a user is called **user profiling** [41]. User profiling has been used to build personalized recommendation system [82], preference detection [127], and personalized chat-bot [88].

Emotion analysis of SR text needs to address three major challenges. Firstly, SR text is mostly written in a casual style in a relatively short form. Thus context information and linguistic cues used are rather limited compared to formal text such as news articles. However, the lack of full text and context information makes it very difficult to use traditional feature engineering methods. Past works primarily rely on additional sentiment lexicons and semantic orientation to improve the task of emotion classification [173, 166]. However, the proposed methods mostly work on formal style text and do not fit well in SR text.

Secondly, cognitive studies concur with linguistics theories that not all words are created equal. Some words are more important than others in conveying semantic meanings. Thus, attention models in neural network are incorporated into emotion analysis to highlight the salient words to convey semantic meanings in a sentence. However, **attention models** built for emotion analysis mostly use information embedded in local context [252, 33]. Using distributional similarity to build attention models lacks theoretical foundation to reflect the cognitive basis. The connection between cognition grounded data and attention mechanism is yet to be directly models, and there is not yet any study on using cognition grounded data to improve sentiment analysis.

Thirdly, by commonsense we know that SR text is more subjective and different comments written by the same person have the tendency to be biased towards personal prefer-

ences. Thus, an intuition is that building an emotion analysis model with explicit user preferences, referred to **user-profile-based models**, can further improve the performance of emotion classification tasks. When building a user-profile based emotion analysis model, we need to address two issues. The first issue is to find an appropriate method to represent user profiles. User profile representation can be learnt from **user activities**. Like many social media data, user activities follow the long-tail distribution [148]. Thus an appropriate data representation model must deal with the data sparseness problem in SR text. The second issue is how to incorporate user profile into emotion analysis models for SR text.

Motivated by these challenges, this thesis studies four areas in subjectivity linked emotion analysis including (1) improving emotion analysis from cognitive perspective by identifying more linguistic features appropriate for our genre of text, (2) using cognition grounded data to improve emotion prediction models, (3) learning the representation of user profiles by addressing the data sparseness through two methods, and (4) incorporating user profiles into emotion analysis model to take subjectivity as a bias into consideration.

Firstly, we propose a linguistics driven model to explore the use of additional morpho-syntactic and orthographic features such as passive construction, verb order as well as some less explored textual features such as unusual use of punctuations and symbol-switches which are often relevant in the causal style of text (social media text). The additional morpho-syntactic features are able to introduce genre and stylist information encoded in the social media text and other casual style text.

Secondly, we explore the use of cognition grounded eye-tracking data to train attention models to improve the performance of linguistics-driven models. In order to bridge the gap between sentiment analysis and cognitive process, we propose a cognition grounded attention model by using eye-tracking data. We first build a regression model to predict reading time of words in sentences based on eye-tracking data. The estimated reading time can then be used as the attention weights in its context to build the attention layer in a neural network based emotion analysis model. Our proposed method also captures

attention more comprehensively using a two level approach where words in their context at both the sentence level and the document level are considered. This part of work validates the effectiveness of cognition grounded data in building attention models, and it proves that the indirect connection between cognition grounded data and sentiment can be modeled to improve sentiment analysis under the attention mechanism.

Thirdly, we explore a novel approach to predict user preferences by learning from both observed comments and missing comments based on the missing-not-at-random hypothesis. To further leverage on social network links available in SR text, we explore methods to extend the context of user profiles through network links. We propose a novel approach to learn node embedding through a joint learning framework of both network links and text associated with nodes. The method can handle both homogeneous networks and heterogeneous networks with multiple types of links. A novel attention mechanism is also proposed to make good use of text extended through links to obtain a larger network context.

Lastly, we propose a deep learning method to incorporate biased user profile into emotion analysis for review text. Individual user bias and preferences, referred to as user profiles, are learned through a neural network model. Then, user profiles as a collection are aggregated by a memory network to encode the user bias. A separate memory network is also used to learn product information. In this way, user profiles and product information can be captured more efficiently as they are by nature different. The final step merges the dual memory networks into a unified classification model for optimization.

The rest of the thesis is organized as follows. Chapter 2 introduces basic concepts and related work. Chapter 3 introduces a linguistics driven model to explore the use of additional morpho-syntactic and orthographic features suitable for SR text. Evaluation shows that our method is very effective for social media text compared to formal news text. Chapter 4 presents our work on using cognition grounded eye-tracking data to train attention models to further improve the performance of the linguistic-driven models. We compare our model with other state-of-the-art attention models using four review datasets

with very convincing result. Chapter 5 introduces our work on learning user profile information. The first part of the work predicts user preferences by considering missing comments, and the second part also extends user context through links in social networks. Comparison to the state-of-the-art models using a number of datasets clearly indicates the advantage of our proposed method. Chapter 6 introduces a deep learning method to incorporate user profiles into an emotion analysis model. The advantages of our proposed model are demonstrated by performance evaluation against other neural network models on three review datasets with user and product information. Chapter 7 concludes the thesis by summarizing the main contributions, current limitations, and future work.

Chapter 2

Background

This chapter introduces background and related works of this thesis. The related works includes three parts. The first part introduces different emotion models and gives an overview of the development of emotion analysis. The second part gives an overview of different user profiling models. The third part reviews related works on using user profiles in emotion analysis.

2.1 Emotion analysis

The term emotion has many different definitions. In general psychology, Scherer et al.[196] give a formal definition of emotions as *episodes of coordinated changes in several components (including at least neurophysiology 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.* Emotion can encoded in text, speech, and facial expression etc. Most works in NLP, including this thesis are focus on emotion in text. As the most complicated and fascinating part of human, emotion has attracted many studies from different aspects including the emotion mechanisms, how to express emotions, and how to recognize emotions [92, 121].

The object of emotion mechanism study is to identify what is emotion and how emotion

generated[1]. Since emotion mechanism is mainly a research topic in psychology and neural science. This thesis will not review the details of emotion mechanism studies.

In addition to understand the emotion mechanism, researchers have also tried to represent the different types of emotions and different emotion models are proposed. The emotion model has been explored from two fundamental view points [94]: The first view point think emotions are discrete and fundamentally different constructs, the second view point think emotions are characterized on dimensional basis in grouping. Emotion can represented by discrete emotion models based on the first view point, and by dimensional emotion models based on the second view point.

Discrete emotion models categorize emotion as a set of independent labels. For example, categorizing emotion by its polarity as positive, negative, and neural is the most straight forward emotion model[103], the term **sentiment** is used to describe this emotion model. In more complicated emotion models, Ortony et al. [169] 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. Ekman et al.[57, 55, 56] identified six basic emotions based on studying the isolated culture of people from the Fori tribe in Papua New Guinea [2]. The tribe members were able to identify these six emotions on the pictures: *Anger, Disgust, Fear, Happiness, Sadness, Surprise*. Despite this study is based on facial expression, it is the most commonly used discrete emotion model in text analysis[121].

Dimensional emotion models represent emotion as a set of values in continuous scales of some multi-dimensional space. Mahranian et al. [150] proposed a Pleasure-Arousal-Dominance (PAD) emotional model. This model makes distinctions of emotion states

[1] https://www.iep.utm.edu/emotion/

[2] https://en.wikipedia.org/wiki/PapuaNewGuinea

by the average values regarding the three dimensions of emotions. This work argues that dimensions of emotions include evaluation (or pleasure), activity and potency (or dominance) which are used to measure stimuli. The evaluation-activity-potency (EPA) model are also called Pleasure-Arousal-Dominance (PAD) model or Valence-Arousal-Dominance (VAD) model [192].

Emotion analysis is the process to recognize emotions represented by emotion models [86]. Generally speaking, emotion analysis can be performed at three levels of granularities the processed text. **Document level emotion analysis** assumes that each document expresses opinions on a single entity. Document level emotion analysis is to determine the overall opinion of the document. Typical data comes from product reviews, and news comments. The second is **at sentence level**. In this level, polarity or emotion labels are learnt for each sentence. Each sentence is considered a separate unit and different sentences can express different opinions. Sentence level datasets are mostly from picked-up sentences in news contents and news headlines. The third level is **feature level emotion analysis**. Feature level emotion analysis identifies an aspect of some products for which the opinion is expressed [108]. This thesis mainly focus on sentence level and document level emotion analysis. Hence we will mainly introduces related works in sentence and document level emotion analysis.

2.1.1 Emotion analysis methods

The development of emotion analysis can be divided into four steps: rule based models, linear based machine learning (ML) models, and deep learning based machine learning models.

Rule based models:

Emotion analysis methods in the earlier years of studies are mainly rule-based either using manually defined rules, lexicons, or linguistic patterns to analyze emotions either at the

sentence level or document level. Wu et al.[244] propose a set of emotion generating rules (EGR) which 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. [31] define a rules set based on the WordNet[3] to analyze the emotions of news headlines. For example, the word inherited from *"Unhealthiness"* in the WordNet will boost *fear* and *sadness* emotions. The authors also define the rule that the main subject in a sentence should weight more. Another type of rule-based methods built based on emotion lexicon. This type of rule-based methods adds the intensity of every words in each emotion category and takes the category with the highest intensity level as the final emotion label [204, 251]. This method is depend on specific lexicon resources such as Sentiwordnet[4] [7], Vader lexicon[5] [59], and Emotion-Potency-Activation (EPA) lexicon[6] [3]. Rule-based methods generally give a high precision. However, one of their drawback is the generalization problem and the time-consuming during the rule definition process.

Linear based machine learning models:

Recent EA methods are more focused on machine learning (ML) models, whose framework is shown in Figure 2.1. This framework shows that ML models 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 (BOW) features, the Part-of-Speech (POS) features, the n-gram features, the lexicon based features [161, 233]. Based on the feature representation, a classifier or a regression model is trained on the

[3] https://wordnet.princeton.edu/

[4] http://sentiwordnet.isti.cnr.it/

[5] https://github.com/cjhutto/vaderSentiment

[6] http://www.indiana.edu/ socpsy/papers/EmotionIdentification.htm

emotion corpus and then used for emotion prediction. In the early stage, most researchers use the linear classifier model with feature engineering. In this period, Supporter Vector Machine (SVM) classifier has achieved great success in text classification [176, 173], with effective feature engineering, SVM was considered one of the best sentiment/emotion classification method before deep learning methods came out. But the feature engineering process still consider as time-consuming and special designed feature templates works for specific dataset only.

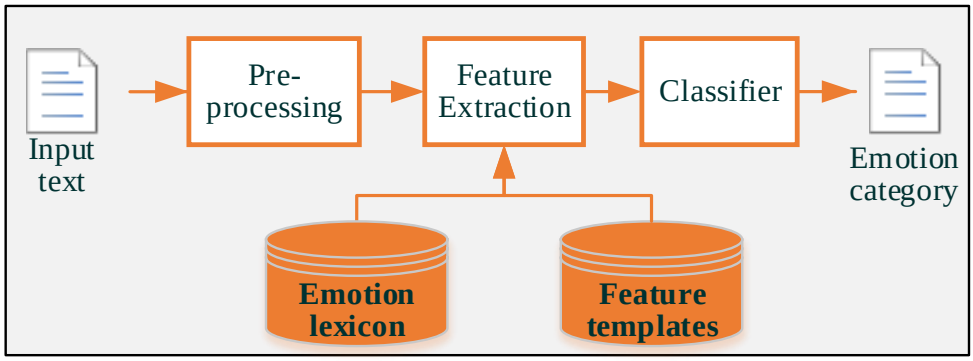


Figure 2.1: Machine learning framework for emotion analysis.

Artificial Neural network:

Deep learning model learning tasks by using multiple layers of artificial neural networks (neural networks for short).

Neural network made up of a large number of information processing units organized in layers. The information processing units are called neurons. The early idea of neural network was inspired from the structure of human brain. In the learning process, the neural network can resembling the learning process of a human brain by adjusting the connection weights between neurons.

Categorized by network topologies, neural network can be further divided into feedforward neural network and recurrent/recursive neural networks. A demonstration of a simple feedforward neural network are shown in Figure 2.2:

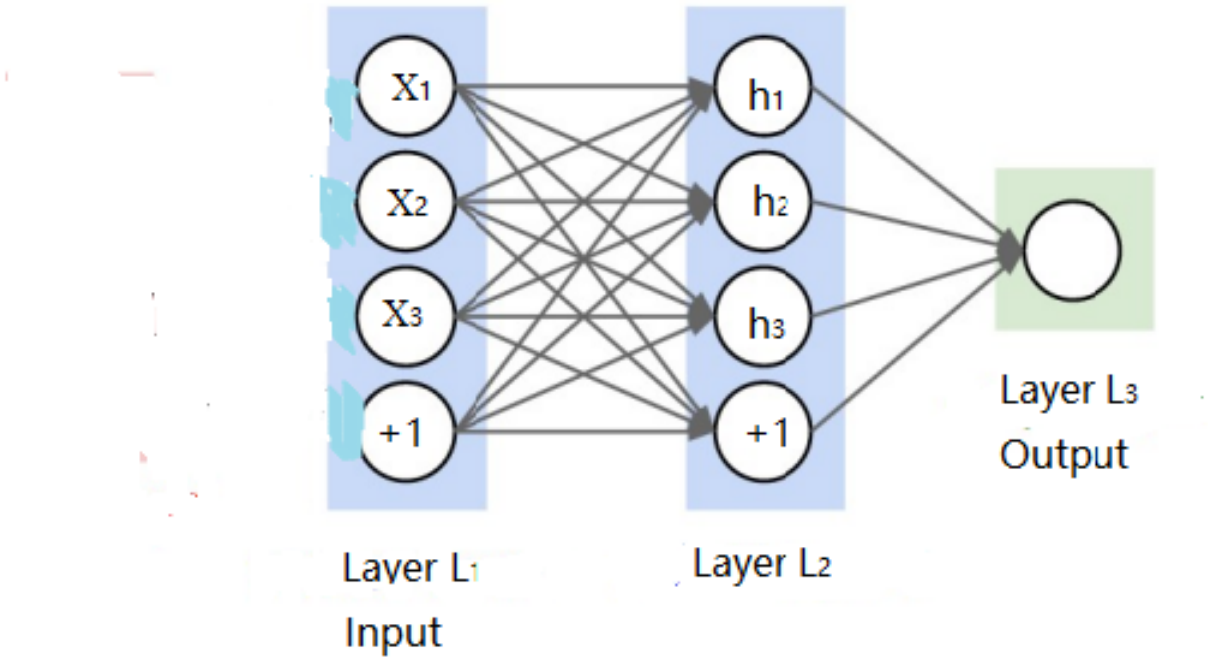


Figure 2.2: Framework for feedforward neural network.

There are three layers in Figure 2.1. L_1 is the input layer, the input layer corresponds to the input vector (X_1, X_2, X_3) and intercept term $+1$. The output layer L_3 made by the output vector and output term. The hidden layer L_2 is not visible as the network output. A circle in L_1 represents an element in the input layer, while a circle in L_2 or L_3 represents a neuron. Neuron is the basic computation element of a neural network. In essence, neuron is an activation function. A line between two neurons represents a connection for information flow. Each connection is associated with a weight, a value controlling the signal between two neurons. The learning of a neural network is achieved by adjusting the weights between neurons with the information flowing through them. Neurons read output from neurons in the previous layer, process the information, and then generate output to neurons in the next layer. As in Figure 2.2, the neural network alters weights based on training examples (x^i, y^i). After the training process, it will obtain a complex form of hypotheses result $h_{w,b}(x)$ that learns from the training data.

For the hidden layer L_2, we can see that each neuron in L_2 takes input x_1, x_2, x_3 and

intercept +1 from L_1, and outputs a value $f(w^t x) = f(\sum_{i=1}^{3} W_i * x_i + b)$ by the activation function f. W_i are weights of the connections; b is the intercept or bias; f is normally non-linear often sigmoid function (sigmoid) [7], hyperbolic tangent function (tanh) [8], or rectified linear function (ReLU) [9]. Their equations are as follows.

We can use the softmax function as the output neuron in L_3, which is a generalization of the logistic function that squashes a K-dimensional vector X of arbitrary real values to a K-dimensional vector $\sigma(X)$ of real values in the range $(0, 1)$ that add up to 1. Generally, softmax is used in the final layer of neural networks for final classification in feed-forward neural networks. Other final output functions include sigmoid, tanh, softmax, ReLU, and Leaky ReLU etc.

By connecting together all neurons, the neural network in Figure 1 has parameters $(W, b) = (W^1, b^1, W^2, b^2)$, where $W^{(l)}_{(i,j)}$ denotes the weight associated with the connection between neuron j in layer l, and neuron i in layer $l + 1$. ${b_{(i)}}^l$ is the bias associated with neuron i in layer $l + 1$.

To train a neural network, stochastic gradient descent via backpropagation is usually employed to minimize the cross-entropy loss, which is a loss function for softmax output. Gradients of the loss function with respect to weights from the last hidden layer to the output layer are first calculated, and then gradients of the expressions with respect to weights between upper network layers are calculated recursively by applying the chain rule in a backward manner. With those gradients, the weights between layers are adjusted accordingly. It is an iterative refinement process until certain stopping criteria are met.

[7] $https://en.wikipedia.org/wiki/Sigmoid_function$

[8] $https://en.wikipedia.org/wiki/Hyperbolic_function$

[9] $https://en.wikipedia.org/wiki/Rectifier_(neural_networks)$

Deep learning (Deep neural network):

Deep learning essentially is multiple layers of "deep" neural network. Motivated by the successful utilization of deep neural networks (DNN) in computer vision [38], speech recognition [43] and natural language processing (NLP) [16], deep neural network based emotion analysis models are proposed to learn low-dimensional text features from end-to-end [201, 203]. Most proposed neural network models take the text information in a sentence or a document as input and generate the semantic representations using well-designed neural networks.

Word embedding

: Many deep learning models in NLP need word embedding results as input features [255]. Note that word embedding is only one to represent a word. 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 [121]. Word embedding belongs to distributional representation. The hypothesis behind word embedding is the word occur in similar context will have similar meaning. The similar meaning is encoded by a low-dimensional dense vector representation. Mikolov et al. [154] propose two widely-used word embedding models called the CBOW model (Continuous Bag-of-Words Model) and the Skip-gram model. The CBOW model predicts the target words from its context words, while the skip-gram model does the opposite, predicting the context words given the target word. According to Zhang et al[255], the CBOW model trend to capture a great deal of information for smaller datasets and the skip-gram model is better for larger datasets. Another frequently used word embedding model is Glove (Globel Vector), which is trained on the non-zero entries of a global word co-location matrix.

Convolutional neural network

Convolutional neural network (short as CNNs) is a special type of neural network initially widely used in Computer vision research. CNNs are inspired by how mammals perceive the world visually [85]. In CNNs, the layers are organised in 3 dimensions: width, height and depth. Further, the neurons in one previous layer do not connect to all the neurons in the next layer but only to a small region of the next layer. Lastly, the final output will be reduced to a single vector of probability scores, organized along the depth dimension.

Typically CNNs have two components:

- The Hidden layers/Feature extraction part: In this part, the network will perform a series of convolutions and pooling operations during which the features are detected. If you had a picture of a zebra, this is the part where the network would recognise its stripes, two ears, and four legs.

- The Classification part: In this part, the fully connected layers will serve as a classifier on top of these extracted features. They will assign a probability for the object on the image being what the algorithm predicts it is.

Recurrent neural network and LSTM network:

Unlinke CNNs, Recurrent Neural Network (RNNs) is a class of artificial neural network where connections between nodes form a directed graph along a sequence [10]. In RNNs, the second layer is called the hidden layer and each node is a hidden node. The framework of Recurrent Neural Network (RNN) is demonstrated in Figure 2.3.

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

$$\vec{h_i} = \sigma(V\vec{h}_{i-1} + U\vec{v}_i + b) \tag{2.1}$$

[10] $https://en.wikipedia.org/wiki/Recurrent_neural_network$

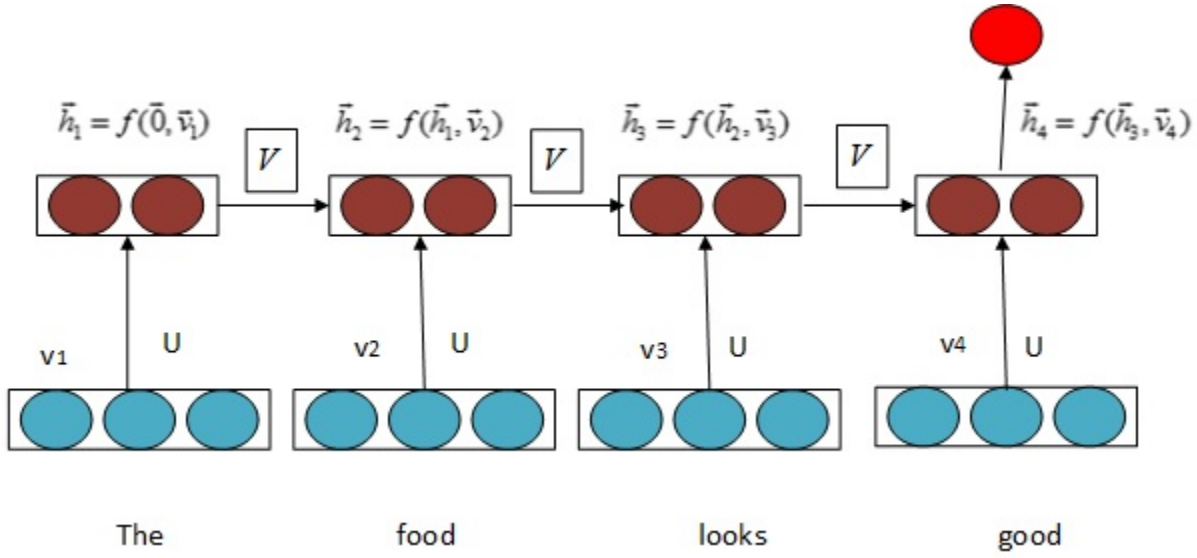


Figure 2.3: Framework for Recurrent neural network.

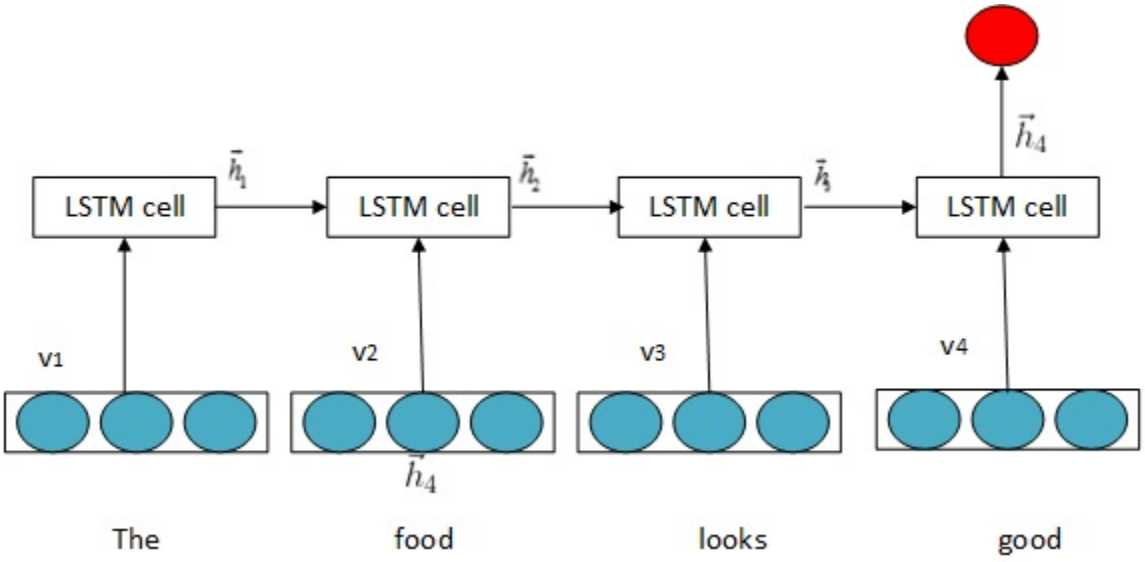


Figure 2.4: Framework for Long Short-Term Memory (LSTM) network

where h_i is the representation of the hidden node i, which is composed from h_{i-1} and the current input node v_i. However, a drawback with RNN is that it suffers from the gradient vanishing (or exploding) problem when the sequence is long (like long sentence in NLP tasks). To overcome this problem, more complex composition model is proposed.

One of the most widely used model is Long Short-Term Memory (LSTM) network. Figure 2.4 shows the general framework of LSTM. In LSTM, the function f is a group of different functions rather than a simple function. Similar to RNN, the box in Figure 2.4 is called LSTM cell is composed from h_{t-1} and the current input node t_i. But an LSTM cell at position t consists of four parts: an input gate vector i_t, a forget gate vector f_t, an output gate vector o_t, and a cell state vector c_t. The output of each LSTM cell is defined by an output vector h_t. These components are defined as:

$$\vec{i_t} = \sigma(U_i\vec{x}_t + W_i\vec{h}_{t-1} + \vec{b_i}), \tag{2.2}$$

$$\vec{f_t} = \sigma(U_f\vec{h}_t + W_f\vec{h}_{t-1} + \vec{b_f}), \tag{2.3}$$

$$\vec{o_t} = \sigma(U_o\vec{h}_t + W_o\vec{h}_{t-1} + \vec{b_o}), \tag{2.4}$$

$$\vec{o_t} = \sigma(U_c\vec{h}_t + W_o\vec{h}_{t-1} + \vec{b_o}), \tag{2.5}$$

$$\vec{c_t} = f_t\vec{c_{t-1}} + \vec{i_t} * tanh(U_c\vec{h}_t + W_c\vec{h}_{t-1} + \vec{b_c}), \tag{2.6}$$

$$\vec{h_t} = \vec{o_t} * tanh\vec{c_t}, \tag{2.7}$$

where σ is the sigmoid activation function, $\vec{b_i}, \vec{b_f}, \vec{b_o}, \vec{b_c}$ are the bias vector. U_i, U_f, U_o, U_c are the model optimization matrix parameters that are learned during training the model. LSTM is good at remembering values for either long or short durations of time.

Attention mechanism:

LSTM model generally been regarded as have ability to keep long time memory. But another problem is LSTM model treat individual component in the input in a equal weight

manner. To overcome this problem, a recent trend in deep learning are Attention Mechanisms. Attention mechanism are built because not all words contribute equally to the representation of the sentence meaning. In a neural network, attention mechanism aims to extract such words that are important to the meaning of the sentence and aggregate the representation of those informative words to form a sentence vector. Let us suppose word i in sentence contains T_i words. with $t \in [1, T]$ represents the words in the i th sentence. The attention mechanism are represented as following functions:

$$\vec{u}_{it} = tanh(W_w \vec{h}_{it} + \vec{b}_w), \tag{2.8}$$

$$\alpha_{it} = \frac{exp(\vec{u_{it}})u_w}{\sum_{i=1}^{t} exp(\vec{u_{it}})\vec{u}_w}, \tag{2.9}$$

$$\vec{s}_i = \sum_{i=1}^{t} \alpha_{it} \vec{h}_{it} \tag{2.10}$$

In this three functions, the attention model first transfer the word representation $\vec{h}_{it}$ obtained through the previous layer to get $\vec{u}_{it}$, then the attention model measure the importance of the word as the similarity of $\vec{u}_{it}$ with a word level context vector $\vec{u}_w$ and get a normalized importance weight α_{it} through a soft-max function. After that, we compute the sentence vector $\vec{s}_i$ as a weighted sum of the word annotations based on the weights. The context vector $\vec{u}_w$ can be seen as a high level representation of additional resources. The additional resources can be a query [209], a global context [252], or additional user/product information [33].

Memory network:

Memory networks (MenNN) have several inference components combined with a large long-term memory in order to learn the memory in a global profile level. The concept of

MemNN was introduced by Weston et.al [239] for solving the question answering problem. The inference component can be neural network and memory acts as a dynamic knowledge base. A MemNN model is built by a memory m (essentially an array of objects) and four other major components. Given an input x, the **Input Feature Map Component**, denoted by I, first converts x to the needed internal feature representation I. x can be an input word, a sentence or a document depending on object of interest. Then, the **Generalization Component**, denoted by G, updates the old memories m_i given the new input. The simplest form of G only process with $I(x)$, $m = G(I(x))$. More sophisticated variants of G can go back and update earlier stored memories or all memories based on the new evidence from the current input $I(x)$. The process is called generalization as there is an opportunity for the network to compress and generalize its memories at this stage for some intended future use. The **Output Feature Map Component**, denoted by O, produces a new output o (in the feature representation space), given the input and the updated m $o = O(I(x), m)$. The **Response Component**, denoted by R, decodes the output o to give the final response: $r = R(o)$. r can be a text response, an action, or a classification label.

Sukhbaatar et al.[209] further modify the MenNN into a End-To-End fashion (MemN2N). This work demonstrates that multiple computational hops in the O component can uncover more abstractive evidences than a single hop and yield improved results for question answering and language modeling. It is worth noting that each computational layer can be a content-based attention model. MemN2N can be regarded as a refined version of the attention mechanism to some extent.

Deep learning based sentiment analysis:

Deep learning model are widely used in sentiment analysis tasks. Commonly used models include Convolutional Neural Networks (CNN)[202], Recursive Neural Network (ReNN) [203], and Recurrent Neural Networks (RNN) [89]. In many works([89, 8, 195]), RNN has

been proven to naturally benefits sentiment classification because of its ability to capture sequential information in text. However, standard RNN suffers from the gradient vanishing problem [17] where gradients may grow or decay exponentially over long sequences. To address this problem, Long-Short Term Memory model (LSTM) is introduced by adding a gated mechanism to keep long term memory. Each LSTM layer is generally followed by mean pooling and then feed into the next layer. Experiments in datasets which contain long documents and sentences demonstrate that the LSTM model outperforms the traditional RNN [214, 215].

Not all words contribute equally to the emotion expression of a sentence [71]. For example, emotion trigger words such as many function words and constructions can shift the emotion polarities of a piece of text [124], emotion carrier words can express emotion directly [104]. Attention mechanisms in neural networks are proposed to highlight their difference in contribution [252]. In document level emotion classification, both sentence level attention and document level attention are proposed. In the sentence level attention layer, an attention mechanism identifies words that are important. Those informative words are aggregated as attention weights to form sentence embedding representation. This method is generally called local context attention method [33]. Similarly, some sentences can also be highlighted to indicate their importance in a document. The local attention mechanism can enforce the model to attend to the important part of a sentence.

However, external knowledge bases such as linguistic knowledge and other aspects of cognition grounded resources, have not been fully employed in neural network models. To this problem, Qian et al.[183] attempts to fully employing linguistic resources to benefit emotion classification. Three types of resources are addressed in Qian's work: sentiment lexicon, negation words, and intensity words. Sentiment lexicon offers the prior polarity of a word which can be useful in determining the sentiment polarity of longer texts such as phrases and sentences. Negators are typical sentiment shifters which constantly change the polarity of sentiment expression. However, the result of this model still under-perform

Name	Paper	Size	Schema	Language
GI	[207]	3,626	Sentiment	English
MPQA	[59]	4,459	Sentiment	English
VADER	[59]	7,502	Sentiment	English
ANTUSD	[234]	27,221	Sentiment	Chinese
DULTIR	[246]	10,259	Emotion	Chinese
ANEW	[22]	1,034	VAD	English
CVAW	[253]	1,647	VAD	English
DAL	[240]	8743	VAD	English
ANGST	[152]	1,003	VAD	German
EPA	[78]	4,505	EPA	English

Table 2.1: A selection of emotion/sentiment lexicons

the state-of-the-art baseline which did not use any external knowledge base.

2.1.2 Emotion analysis resources

Adopting external resources are proven to useful for emotion analysis[164, 59]. The emotion analysis resources related to this thesis include emotion lexicons and cognition grounded resources.

Emotion lexicons are important resources for emotion analysis. These emotion lexicons consist of predetermined list of words assigned to emotion label or values, which is baseline for many machine learning based methods[131, 212]. Depending on emotion models, there are two mainstream labeling schema, the first schema is representing affective meanings of words by discrete emotion labels, such as *positive, negative*, or *happiness, sadness, anger etc* [57]. The second schema is to represent affective meanings by the more comprehensive multi-dimensional representation models like the valence-arousal-dominance model (VAD) [192] and the evaluation-potency-activity model (EPA) [77].

Emotion lexicons are obtained either by **manual annotation** or **automatic methods**. Manually annotated sentiment lexicons include the General Inquirer (GI) [207], MPQA

[188], VADER [59], ANTUSD [232] in Chinese. General Inquirer is the first sentiment lexicon in English, labeled though questionnaires with 3,626 words. General Inquirer (GI) is the first sentiment lexicons in English which contain 3,626 words. MPQA is labeled from approximately 700 documents with 4,459 words from news text. VADER contains 7,502 words annotated from twitter. ANTUSD is constructed by collecting sentiment stats of 27,221 Chinese words in several sentiment annotation work. Manually annotated emotion lexicons based on discrete emotion models include the DULTIR emotion lexicon in Chinese [246], the English emotion lexicon [165] which contains about 17,000 words.

Manually annotated multi-dimensional lexicons in other affective dimension include ANEW, CVAW, DAL, EPA and ANGST, etc. The ANEW lexicon is based on a three dimension model on Valence, Arousal, and Dominance (VAD) model [22] which contains 1,034 English words. Valence can directly serve as the sentiment dimension. The extended ANEW lexicon contains about 13,965 English words annotated through crowd-sourcing. The CVAW lexicon based on the VAD model [253] contains 1,653 traditional Chinese words in the valence and arousal dimensions. The Dictionary of Affect in Language (DAL) lexicon annotated in the dimensions of Pleasantness-Activation-Imagery contains 8,742 terms [240]. Pleasantness can directly serve as the sentiment dimension. The Evaluation, Potency, and Activity (EPA) lexicon annotated in the evaluation-potency-activity schema [78] contains about 4,505 English terms. Here the evaluation dimension is close to sentiment in the EPA schema. The ANGST lexicon annotated in the valence-arousal-potency dimensions contains 1,003 German words [198]. But the biggest problem for manual annotation is high costs in both time and resources. Hence most of manually annotated resources is limited in size. That problem is especially serious in neural network models.

Given the limitation of manually labeled resources, researches start to apply *automatic methods* to build lexicon. Automatic methods to obtain emotion lexicons are focused mainly on the sentiment space because current research works are mostly on sentiment analysis [130, 27, 73, 123]. In terms of methodology, there are mainly three approaches to

build lexicons automatically.

The first approach of automatic methods uses statistical information between a target word and seed words. The seed words are manually labeled or borrowed from other existing emotion lexicons. For example, sentiment polarity intensities are calculated based on Point-wise Mutual Information (PMI) between a target word and positive seed words or negative seed words, respectively [228, 164]. Similarly, PMI is used to build discrete emotion lexicon based on naturally annotated hash-tags in twitter[163].

The second approach of automatic methods is based on label propagation method. The label propagation method firstly builds a word graph and then label propagation is performed to infer the affective values of unseen words from the seed words. For example, a graph can be built based on the semantic relationship in WordNet[11] and the label propagation is performed to infer the EPA values [3] and emotion polarity[194]. A knowledge based graph is confined by the coverage of a knowledge base. A word graph can also be built from a text corpus based on cosine similarity of words represented by their contexts words and then graph propagation is performed to infer the sentiment polarity of unseen words [229]. Word embedding is also used to compute cosine similarity between words to build the word graph [254]. Similarly, a word graph is constructed using cosine similarity of word embedding to infer sentiment polarities [73].

The third approach represents a word as a dense vector and then map this vector to some sentiment value or categories based on a regression model or a classifier. Features used in vectors can be either manual defined or by expert knowledge. Then features are processed by linear regression to obtain sentiment labels or scores [238, 217]. A recent work proposed by Li et al.[123] introduce a ridge regression model to inferring affective meanings of words from word embedding. Evaluation on various affective lexicons shows that ridge regression outperforms the state-of-the-art methods on all the lexicons under different evaluation metrics with large margins. Fellowing the works conducted by Li et

[11] https://wordnet.princeton.edu/

al. [123], automatic methods can easily expand its scale, hence we did not show the size of each automatically built lexical.

Another important resource can help to improve emotion analysis is cognition grounded data. Eye-tracking data is one of the commonly used cognition grounded data [18]. In the simplest terms, eye-tracking measures eye activity. Eye-tracking data is collected using either a remote or head-mounted tracker device connected to a computer.

Among different available eye-tracking datasets, the Dundee corpus[101], GECO (Ghent Eye-Tracking Corpus)[40], and Mishra et al. [159] are considered as three high-quality resources. The Dundee corpus contains eye movement data from English and French newspapers [101]. Measurements are taken while 10 participants read 20 newspaper articles. GECO is an English-Dutch bilingual corpus with eye-tracking data from 17 participants collected from reading the complete novel *The Mysterious Affair at Styles*. The corpus has 4,934 sentences, 774,015 tokens, and 9,876 words. The Mishra et al. [159] dataset contains 994 text snippets with 383 positive and 611 negative examples from newspaper clippings, sampled from seven native speakers.

2.1.3 Emotion analysis datasets

As an emotion dataset is the premise for emotion analysis model, emotion dataset construction becomes another heated research topic in emotion analysis community. Emotion dataset construction are generally regarded as a difficult task. Because obtaining labeled emotion data can be very time-consuming and noise prone especially for the subjectivity related emotion label annotation.

Based on the approach to obtain emotion labels, emotion dataset construction methods can divided into two categories: the first category is manual annotation by experts or crowd-sourcing. There are some emotion corpora based on manual annotation. For English, Strapparava et al. [208] provide "Affective Text" task dataset focuses on the classification of emotions and valence (emotion polarity) in news headlines. This dataset contains

1,250 news headlines labeled with the six Ekman emotion labels [57]. Scherer et al. [197] provide a dataset which consist of 7,666 sentences generated through questionnaires. As for social media text, Yan et al.[248] construct a gold standard corpus of 15,553 tweets annotated with 28 emotion categories for the purpose of training and evaluating machine learning models for emotion classification through Amazon Mechanical Turk (AMT)[12].

For Chinese, Quan et al.[184] provide a Chinese dataset contain 500 documents, with 4,004 paragraphs, 12,742 sentences, and 324,571 words by manually annotation. Yu et al.[253] build an affective corpus called Chinese valence-arousal text (CVAT) containing 2,009 sentences extracted from web texts. Li et al.[120] build a Chinese Sentiment Treebank over movie reviews data. This dataset includes 13,550 labeled sentences and organized in dependency tree structure. Xu et al.[247] build a Chinese news dataset from Sina News channel. In this work, 8,802 articles with 1,454,912 emotional votes are obtained in total. There are about 165 votes for each article on the average.

The second category aims to build emotion datasets though semi-automatic or automatic methods. The automatic method mainly used in social media, where the text generally contain natural labels, like hash-tag, smileys, and point-based ratings. Davidov et al.[45] built sentiment corpus by utilizing 50 Twitter tags and 15 smileys as sentiment labels. For semi-automatic methods, Li et al. [122] designs a three-stage semi-automatic method to construct an emotion corpus from micro-blogs. Firstly, a lexicon based voting approach is used to verify the hash-tag automatically. Then, a SVM based classifier is used to select the data whose natural labels are consistent with the predicted labels. In the last step, the remain parts are labeled by professional annotators.

Both manually tagging and automatic tagging method have its advantages and disadvantages. The manually label approach generally have higher quality than automatic or semi-automatic methods. But the intensive manual annotation process would limit the scale of corpus. While the automatic or semi-automatic method avoids the need for labor

[12] https://www.mturk.com/

intensive manual annotation, allowing identification and classification of diverse sentiment types of texts. However, the automatic method will bring noise to the corpus [12]. For improving the first methods, obtaining user generated data in review platforms are becoming popular. In the review platform, Human labeled review ratings are regarded as gold standard emotion labels. When the human labeled reviews, they also leave their textual comments for certain products. This method have four aspects of advantages: Firstly, we do not need to manually annotate the emotion labels of text since the users (customers) submit both text content and emotion rating at the same time. Secondly, the huge number of users gathered by review platform made it possible to obtain large scale dataset for sentiment/emotion analysis. Thirdly, the automatic filtering methods provided by the review platform can make sure the text is basically clean. Big review platforms like Yelp. IMDB, and Trip-advisor have developed their own models to filter reviews from not real people as spam[13].Last but not least, the user profile and product information included in the dataset given by the review platform made it possible to take user profile and product information as subjectivity into consideration for emotion analysis. The datasets collected in this approach include [140, 49] and Yelp dataset challenge dataset (Yelp 13, Yelp 14[14]).

The classic review datasets provided by the review platform contain three major components, users (the subjects of behaviors) and products (the objects of behaviors) are regarded as two common types of nodes. User express their emotion towards product though comments. The user profile and product information can gathered from related texts or archived information. For convenience, we use term user to represent the subjects of behaviors and products to represent the object of behavior in the remaining part of this thesis. Because reader also provide feedback towards an entity though text comments, in the review texts, user can be regarded as readers. Table 2.2 lists a selection of sentiment/emotion analysis datasets obtained by different methods.

[13] https://vivial.net/blog/how-to-avoid-the-yelp-review-filter-and-get-more-positive-reviews/

[14] https://www.yelp.com/dataset/challenge

Paper	Level	Language	Size	Task	Resources	Method
[208]	Sentence	EN	1,250	SA	news headline	manually
[197]	Sentence	EN	7,666	EA	daily communication	manually
[248]	Sentence	EN	15,553	EA	social network	manually
[184]	Sentence	CN	12,742	EA	social network	manually
[45]	Sentence	EN	65,000	EA/SA	social network	automatic
[122]	Sentence	CN	48,000	SA	social network	semi-automatic
[247]	Document	CN	8,802	EA	news channel	manually
[140]	Document	EN	50,000	SA	reviewer platform	manually
[49]	Document	EN	84,919	SA	reviewer platform	manually
YELP 14	Document	EN	231,163	SA	review platform	manually
YELP 13	Document	EN	78,966	SA	review platform	manually
[247]	Document	CN	8,802	EA	reviewer platform	manually

Table 2.2: Selection of current Emotion/Sentiment analysis datasets

2.2 User profile construction

Presenting user profile using dense vector representation though user activities is the key to build user profiling models. The purpose of user profiling is to get a representation of a certain user based on user archives and user activities. The user representation can be used to infer user preferences or used in other downstream tasks. The theoretical foundation of user profile is the homophily theory. Homophily, often summarized using the moniker "birds of a feather flock together", is the tendency for individuals to seek out and associate with others who share similar attributes (e.g., beliefs, physical features, and activities)[149]. Under the homophily theory individuals with similar attributes cluster together in on-line social networks and have similar behaviors[2].

Generally speaking, user profile organized in two different ways: explicit user profiles and implicit user profiles. At one hand, user profile can expressed explicitly in user's archive page, like Face book[15] or Twitter[16] provide user archive option for user to put profiles like gender,location,and age. On the other hand, although structured user profile

[15] https://www.facebook.com/

[16] https://twitter.com/

can be readily used in any research and applications [80], such structured information is sparsely available. The majority of user preferences are latent information implicitly expressed through the activities they carry out over the social media.

User activities can be divided into two different parts, the first component is user composed content, including text like comments, posts, and personal status, uploaded videos, and pictures. These texts often contain strong evidence about his preferences on events and entities. As a NLP study, this thesis mostly caring about user composed texts. The second group is link information, like connect with friends and followers. The link information forms a user network to connect a user with different entities.

2.2.1 User text embedding

In many social networks, users have a various of activities involved with text information. User generally express their opinion toward their posts, tweets, or user to user private messages. To mining user profile from the textual related activities becoming a heated research topic.

In the early work of user profile study, using text related activities to extract user profiles are generally set as a feature engineering problem. The feature engineering-based models are largely learned from researchers in sociolinguistics. The sociolinguistics has explored the effects of gender, age, social class, religion, education and other speaker attributes in conversational discourse and monologue. In earliest work in this area conduct by Fischer[58] and Labov[111] involved studying morphological and phonological features respectively. Macaulay et al.[143] demonstrated the differences in lexical choice and other linguistic features in discourse conditioned on age, gender, and social class. For example, in speech it is well known that certain utterances like “umm”, “uh-huh”, and back channel responses like laughter and lip smacking are more prevalent among female speakers than their male counterparts. We summarize different set of features used in user profiling papers in table 2.3.

Feature	Description	Paper
K-top words	The k most differentiating words used by each labeled group were included as individual features	[178, 193, 25]
K-top stems.	Plurals and verb forms can weaken the signal obtained from k-top words by causing forms of the same word to be handled as separate words.	[2, 135]
K-top n-grams	In the training data, the k most differentiating bigrams and trigrams were identified for both labels	[185, 25]
K-top co-stems	In prior work, the ends of words (e.g., conjugations, plurals, and possessive marks) were shown to give notable signal about a variety of blog author attributes	[128]
K-top hashtags	Hashtags operate as topic labels. Prior work has shown that the extent to which topics are attributespecific, they can be used for attribute inference.	[39]
K-top mentions	The named entities mentioned in the content.	[37]
Punctuations	A combination of any number of ? and ! (!?!!??!)	[185]
Slang terms	e.g. delish, cozy, yummy, nerdy, and yuck	[4]
Frequency statistics	number of frequency statistics: tweets, mentions, hashtags, links, and retweets per day	[4]

Table 2.3: Features used in previous works for user profile detection

Noted that the features proposed are generally designed for a specific user profile prediction task. Examples include gender [2, 37], age [185], and political tendency ([178, 39]). The main drawback of feature engineering methods are labor intensive and lack of generalizability. First, obtaining useful feature for user profiling require detailed qualitative analysis on user generated texts, which require researchers have a good understanding of user generated texts and theoretical knowledge in sociolinguistics and statistics. Secondly, specific designed feature only works on certain user profiles, for example, Slang terms is a good indicator of user's social class, but not a valid feature for user's age or location.

Different from feature engineering methods, embedding models represent words as

the basic units to operate on, aiming to capture contextual meanings of text from end-to-end. The center theorem for representation learning is that representations for co-location words should be similar in vector spaced while representations for words in different contexts should be separated. A variety of models based on the center theorem are proposed ([154, 179, 118]) to capture better word representations. Embedding learning frameworks could be optimized through language models [154] or matrix factorization techniques on word-to-word co-occurrence matrix [119]. Embedding models for user generated text representation need to consider larger text units beyond words, in fact, user embedding user generated texts can regarded as a paragraph or document embedding model.

Learning embedded representations for larger text unit like documents and paragraph were initially proposed by Le et al. [113]. In this paper the authors present an unsupervised model as an extension of the Word to vector (word2vec) model, noted as Document to Vector (Doc2vec). This model capable of training what the authors refer to as Paragraph Vectors or document vectors. These Paragraph Vectors are trained in two manners, very similar to the way that word vectors are trained. One method, known as Paragraph-Vector Distributed-Memory or PV-DM, averages or concatenates the Paragraph Vector into the context window for all windows of the document. PV-DM incorporates word order while training. The other method, which ignores word order, is very similar to the Continuous Bag-of-Words (CBOW) method for training word vectors. Known as Distributed Bag of Words version of Paragraph Vector (shorten as PV-DBOW), this technique creates Paragraph Vectors by training to predict words within a window of the paragraph. The resulting paragraph vectors (PVs) created by these methods proved to be very effective in downstream tasks, with results outperforming many (supervised) state-of-the-art methods on the Stanford Sentiment Tree-bank [17] IMDB data sets [18]. Despite proposed in document and paragraph embedding initially, it can easily being used to user generated text embed-

[17] https://nlp.stanford.edu/sentiment/treebank.html

[18] https://www.imdb.com/interfaces/

ding by regrading user generated text as document. Based on [113], researchers continue explore unsupervised document embedding algorithm. [35] proposed a document embedding method called Doc2VecC, stands for Document Vectors through Corruption. Like original Doc2vec mode, The Doc2VecC embedding of a document is simply computed by averaging the embeddings of its component words. However, Doc2VecC introduce the corruption mechanism. To decrease computational cost Doc2VecC randomly ignores significant parts of the text and deliberately zeros out dimensions from the document embedding while training. There are other models aimed at retrofit document embedding algorithm based on Doc2vec model, such as Kusner el al.[110],Pagliardini et al.[171].

Apart from word embedding based approach, deep neural network models also has been used to obtain user generated text embedding. Skip-through model[107] is one of the examples to use deep learning to train sentence embeddings in an unsupervised manner.Skip-through is proposed an encoder-decoder model that tries to reconstruct the surrounding sentences of an encoded passage. The encoder is an unidirectional or bidirectional RNN[61]. The output of the encoder is the sentence embedding, which is then used by two single layered RNN decoders to predict and generate the previous and next sentences in the text. In both cases, Gated Recurrent Units (GRU)[36] are used as RNN cells. Other paper use deep neural models include [81, 126].

2.2.2 User network embedding

Despite generated text contain valuable information for user profiling. Other user activities plays a very important role in user profiling. Especially User network. As a part of network representation learning, user network embedding has been proposed as a critical technique for network analysis task. [65] categorize the embedding methods for user networks into three broad categories: (1) Factorization based methods, (2) Random walk based methods, and (3) Deep learning based methods. Note that random walk based methods and deep learning based methods are not mutually exclusive.

Traditional network embedding approaches use matrix factorization. Factorization based algorithms represent the connections between nodes in the form of a matrix and factorize this matrix to obtain the embedding. The matrices used to represent the connections include node adjacency matrix, Laplacian matrix, node transition probability matrix, and Katz similarity matrix, among others [65].

The method to approach matrix factorization is decide by the properties of matrix. If matrix is semidefinite matrix like the Laplacian matrix [5], eigenvalue value methods like Principal Component Analysis (PCA) [93] or Singular-Value Decomposition (SVD) [62] can used. Locally Linear Embedding (LLE) method [191] is the pioneer model to use eigenvalue method for matrix factorization. LLE assumes that every node is a linear combination of its neighbors in the embedding space. Suppose we have user graph G to make a adjacent matrix W, the weight of connection between node i and j is represented as W_{ij}. The embedding of i, Y_i is represented as the linear combination of node j, noted as:

$$Y_i \approx \sum_i W_{ij} Y_j (\forall i \in V) \tag{2.11}$$

The process to obtain embedding can derived as minimizing the following equations:

$$\phi(Y) = \sum_i \left\| Y_i - \sum_i W_{ij} Y_j \right\|^2 . \tag{2.12}$$

Suppose the variance of embedding constrained as $\frac{1}{N} Y^T Y = 1$ and the center of embedding is zero. The above constrained optimization problem in function 2.12 can be reduced to an eigenvalue problem. Other eigenvalue based model include [14, 136, 200, 170].

Another approach is random walk model. Random walk model is proposed partially because matrix factorization are computationally expensive for large scale user data[65].

Two key works in random walk based method are DeepWalk[180] and node2vec[66].

DeepWalk[153] model uses local information obtained from truncated random walks to learn latent representations by treating walks as the equivalent of sentences. The truncated random walks are performed by maximizing the probability of observing the last k nodes and the next k nodes in the random walk centered at node v_i. The loss of single walk is computed by maximizing the log probability $logP_r\left(v_i - k, v_i - 1, v_i + 1, ..., v_i + k|Y_i\right)$ where the $2k + 1$ is the length of random walk. The deepwalk process will generate multiple random walk in size of $2k + 1$, the optimization function works on the sum of log-likelihoods for each random walk.

Node2vec [66] model also preserve higher-order proximity between nodes by maximizing the probability of occurrence of subsequent nodes in fixed length random walks. Different to deep walk model, Node2vec applies biased-random walks that provide a trade-off between breadth-first (BFS) and depth-first (DFS) graph searches. The choose of balance between breadth-first approach and depth-first approach help preserve community structure as well as structural equivalence between nodes [65].

The third approach is deep neural networks based methods for graphs embedding. Two examples in deep learning models are Structural Deep Network Embedding (SDNE) [232] and Deep Neural Networks for Learning Graph Representations (DNGR) [28]. The work in [232] uses a Structural Deep Network Embedding method (SDNE). The paper first proposes a semi-supervised deep model, which has multiple layers of nonlinear functions, thereby being able to capture the highly non-linear network structure. Then SDNE propose to exploit the first-order and second-order proximity jointly to preserve the network structure. The second-order proximity is used by the unsupervised component to capture the global network structure. While the first-order proximity is used as the supervised information in the supervised component to preserve the local network structure. By jointly optimizing them in the semi-supervised deep model, SDNE can preserve both the local and global network structure and is robust to sparse networks.

Deep Neural Networks for Learning Graph Representations (DNGR) [28] combine neural network with matrix factorization methods. The model consists of 3 components: random surfing, positive point-wise mutual information (PPMI) calculation and stacked denoising autoencoders. Random surfing model is used on the input graph to generate a probabilistic co-occurrence matrix, analogous to similarity matrix. The matrix is transformed to a PPMI matrix and input into a stacked denoising autoencoder to obtain the embedding. Inputting PPMI matrix ensures that the autoencoder model can capture higher order proximity. Furthermore, stacked denoising autoencoders used to aid the robustness of the model in presence of noise in the graph as well as in capturing underlying structure required for tasks such as link prediction and node classification.

Both SDNE and DNGR models take the global neighborhood of each node as input. That can be very computation expensive and not addressed with sparse data. The paper [106] proposed Graph Convolutional Networks (GCNs) to tackle this problem by defining a convolution operator on graph. The model iteratively aggregates the embeddings of neighbors for a node and uses a function of the obtained embedding and its embedding at previous iteration to obtain the new embedding. Aggregating embedding of only local neighborhood makes it scalable and multiple iterations allows the learned embedding of a node to characterize global neighborhood. Other models also using convolution on graphs to obtain semi-supervised embedding, such as [23, 79].

2.2.3 Incorporating multi-types of user information

In contrast to homogeneous networks, a heterogeneous network has multiple types of nodes such as users and videos and multiple types of information associated with the nodes such as text, attributes, and multi-media content. However, most of these network embedding models only encode the structural information into node embeddings, without considering heterogeneous information accompanied with nodes in real-world social networks [224]. To address this issue, Yang et al. [249] present text-associated DeepWalk

(TADW) to improve matrix factorization based DeepWalk with text information.

TADW learn from the fact that Deepwalk model can transform into a matrix factorization model. The Deepwalk process actually equal to factorize matrix $M \in R^{|V|*|V|}$, where each entry M_{ij} is logarithm of the average probability that vertex v_i randomly walks to vertex v_j in fixed steps. Based on matrix factorization transform of Deepwalk, TADW factorize matrix M into the product of three matrices: weighted matrix $W \in R^{k*|V|}$, network features $H \in R^{k*f_t}$, and text features $T \in R^{f_t*|V|}$. Then we concatenate W and HT as 2k-dimensional representations of nodes.

Complex methods, such as the Community-enhanced Network Representation (CENE) [225] leverages both network link information and text information by modeling text as a special kind of nodes, and then optimizes the probabilities of heterogeneous links. Tu et al. [224] proposes a state-of-the-art Context Aware Network Embedding (CANE) model to extract context information with an attention mechanism for text embedding. But CANE was proposed for a homogeneous network. For heterogeneous networks having multiple types of nodes, Gui et al. [67] used a large-scale network embedding model initially proposed by Tang et al.[219] to explore user and product representations. However, when text information is included, comments written by the same user at different times, or comments made by different users of the same product node are treated as isolated text units. Even though individual comments can be short, a collection of them as a document to each node, can give more comprehensive information of the nodes. There are yet methods to explore the use of document information in text embedding for the learning of network embedding. Tu et al. [226] propose max-margin DeepWalk (MMDW) to learn discriminative network representations by utilizing labeling information of vertices. MMDW is a unified Network representation learning (NRL) framework that jointly optimizes the max-margin classifier and the aimed social representation learning model. Based on Deepwalk model, MMDW firstly learns DeepWalk as matrix factorization. Afterwards, it trains a max-margin based classifier and enlarges the distance between support vectors and classi-

fication boundaries. Sun et al. [210] regard text content as a special kind of vertices, and propose context-enhanced network embedding (CENE) through leveraging both structural and textural information to learn network embeddings.

Tu et al. [224] propose a model called CANE (Context-Aware Network Embedding) to learn context-aware vertex embeddings. Tu et al's work categorize network embedding into two categories: context-free embedding and context aware embedding. In conventional network embedding models, each vertex is represented as a static embedding vector, denoted as context-free embedding. CANE assigns dynamic embeddings to a vertex according to different neighbors it interacts with, named as context-aware embedding. Take a vertex u and its neighbor node v for example. The context-free embedding of u remains unchanged when interacting with different neighbors. On the contrary, the context-aware embedding of u is dynamic when confronting different neighbors. In order to realize context-aware text-based embeddings, CANE model introduce the selective attention scheme and build mutual attention between u and v into these neural models. The mutual attention is expected to guide neural models to emphasize those words that are focused by its neighbor nodes and eventually obtain context aware embeddings.

2.3 User profile based emotion analysis

Emotion as a cognitive process is largely subjective and user bias plays a significant role in emotion analysis. Lenient users tend to give higher ratings than finicky ones even if they review the same products with similar wording. However, most existing emotion classification models ignore the biased user subjectivity, which have crucial effects on the emotion polarities. Compare to the works in the conventional emotion analysis, user profile based emotion analysis still in its infant stage. This subsection will introduce the current methods in incorporating user profile information to build a personalized emotion analysis model.

User profile based emotion analysis started with analyzing data from the social network platform like twitter. Tan et al. [213] propose to improve sentiment analysis by utilizing the information about user-user relationships made evident by on-line social networks. The user-user relationships and user-tweet sentiment are jointly modeled based on a factor-graph model to make the user-level sentiment analysis.

Despite twitter provide rich user information for user profile based emotion analysis, user profile based emotion analysis model are more easily applied to review datasets. As we introduced in section 2.1.3, the review datasets generally have three components: the **review** text, the user who posts the review, and the product which is evaluated. Firstly, review dataset are subjective whereas lenient user will have different review rating than finicky users even with same or similar wording. Secondly, review dataset contain rich user and product information, including user generated texts and other network behavior of users. Thirdly, supported by the industry which need an improved user profile based system like Yelp [19], IMDB[20], and Openrice [21], reviewer dataset can have a large scale with user labeled rating.

Tang et al.[214] propose a new model dubbed **User Product Neural Network** (UPNN) to capture user-level and product-level information for sentiment classification of documents (e.g. reviews). UPNN takes as input a variable-sized document as well as the user who writes the review and the product which is evaluated. It outputs sentiment polarity label of a document. Users and products are encoded in continuous vector spaces, the representations of which capture important global clues such as user preferences and product qualities. These representations are further integrated with continuous text representation in a unified neural framework for sentiment classification.

An illustration of UPNN is given in Figure 2.5. It takes as input a review, the user

[19] https://www.yelp.com/

[20] http://www.imdb.com/

[21] https://www.openrice.com/en/hongkong

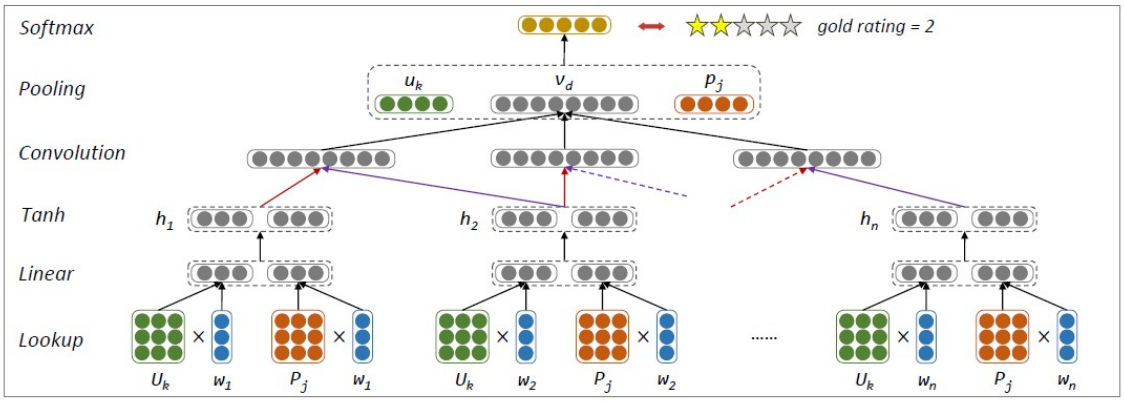


Figure 2.5: An illustration of User Product Neural Network (UPNN)

who posts the review, and the product which is evaluated. u_k and p_j are continuous vector representations of user k and product j for capturing user-sentiment and product-sentiment consistencies. U_k and P_j are continuous matrix representations of user k and product j for capturing user-text and product-text consistencies. The vector representation and matrix representations are feed into Convolution Neural Network (CNNs) for producing user and product enhanced document representation.

While Tang et al. [214] integrate user profile and product information with text information in the Lookup layer (see Figure 2.5). Gui et al. [67] proposed a **user inter-subjectivity network (shorten as UserInter)** which links review writers (users), terms they used, as well as the polarities of the terms. The first step of this model is construct an inter-subjectivity network which links review writers, terms they used, as well as the polarities of the terms. The output of inter-subjectivity network is the representation of users. The representation of users are subsequently incorporated with text representation and then feed into the max-pooling layer of a CNNs for sentiment analysis. But theoretically speaking, the representation of users can be embedded into any sentiment classification

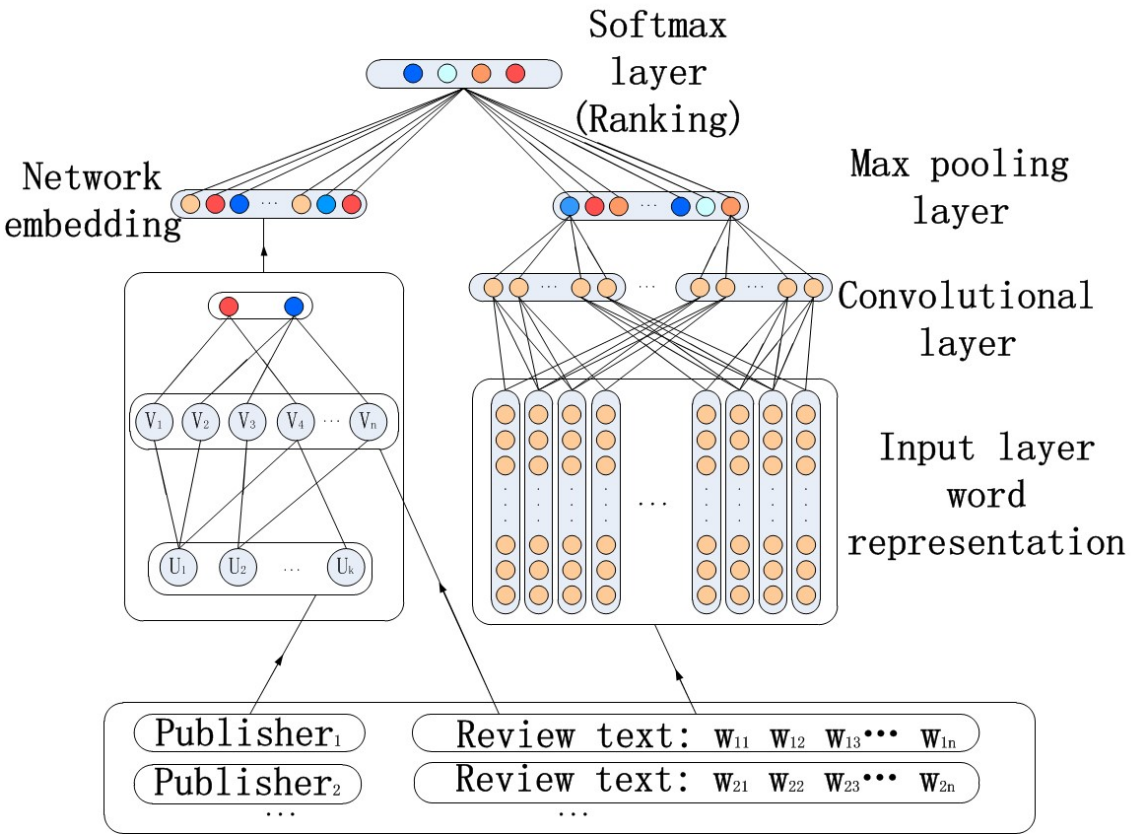


Figure 2.6: An illustration of User subjectivity Network (Inter-subjectivity)

methods. The illustration of user subjectivity network is in Figure 2.6.

Chen et al. [33] provide another approach by proposed a **user attention mechanism** and a **product attention mechanism** into LSTM (Long-short time memory network), noted as LSTM+UPA model. First, LSTM+UPA builds a hierarchical LSTM model to generate sentence and document representations. Afterwards, user profile and product information is considered via attentions over different semantic levels due to its ability of capturing crucial semantic components. The illustration of LSTM+UPA model is in Figure 2.7.

Dou [52] proposes a memory network for document-level sentiment classification (shorten as UMN) which could capture the user and product information at the same time. Based on memory network and Long Short-Term Memory (LSTM), the model can be

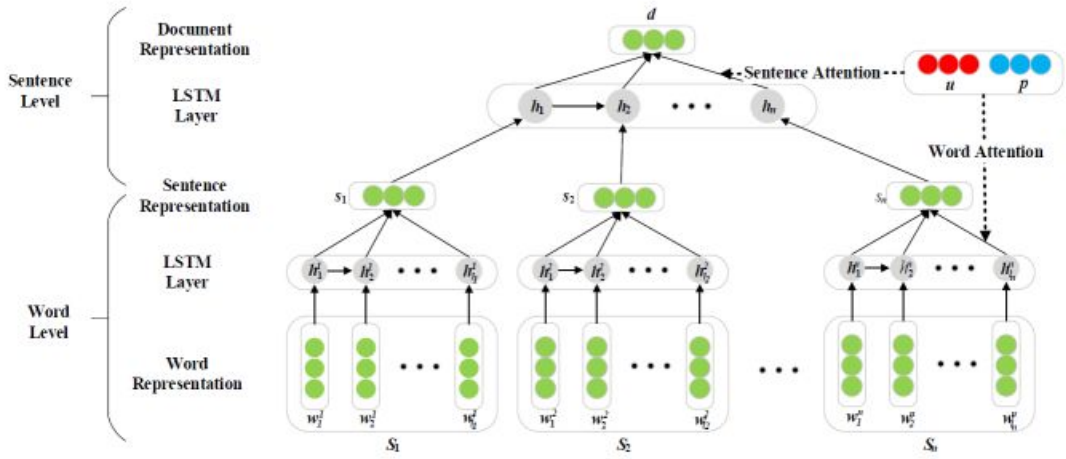


Figure 2.7: An illustration of LSTM+CBA network

Model	Paper	IMDB	Yelp13	Yelp14
UPNN	[214]	0.435	0.608	0.596
UserInter	[67]	0.476	0.623	0.610
LSTM+UPA	[33]	0.532	0.650	0.667
UMN	[52]	0.465	0.609	0.639

Table 2.4: Result comparison (Accuracy) of user profile based sentiment analysis model

divided into two separate parts. In the first part, the proposed work utilizes LSTM to represent each document. Afterwards, we apply memory network consists of multiple computational layers to predict the ratings for each document and each layer is a content-based attention model. The illustration of user memory network is illustrated in Figure 2.8.

The models we mentioned above all conduct experiments in IMDB and Yelp dataset. For comparison, Table 2.4 list performance of models which incorporate user and product information in three benchmark datasets: IMDB, Yelp13, and Yelp 14.

Nearly all currently proposed user profile based emotion analysis model handle user profile and product information in a unified model which may not be able to learn salient features of users and products effectively. By common sense, we know that review text

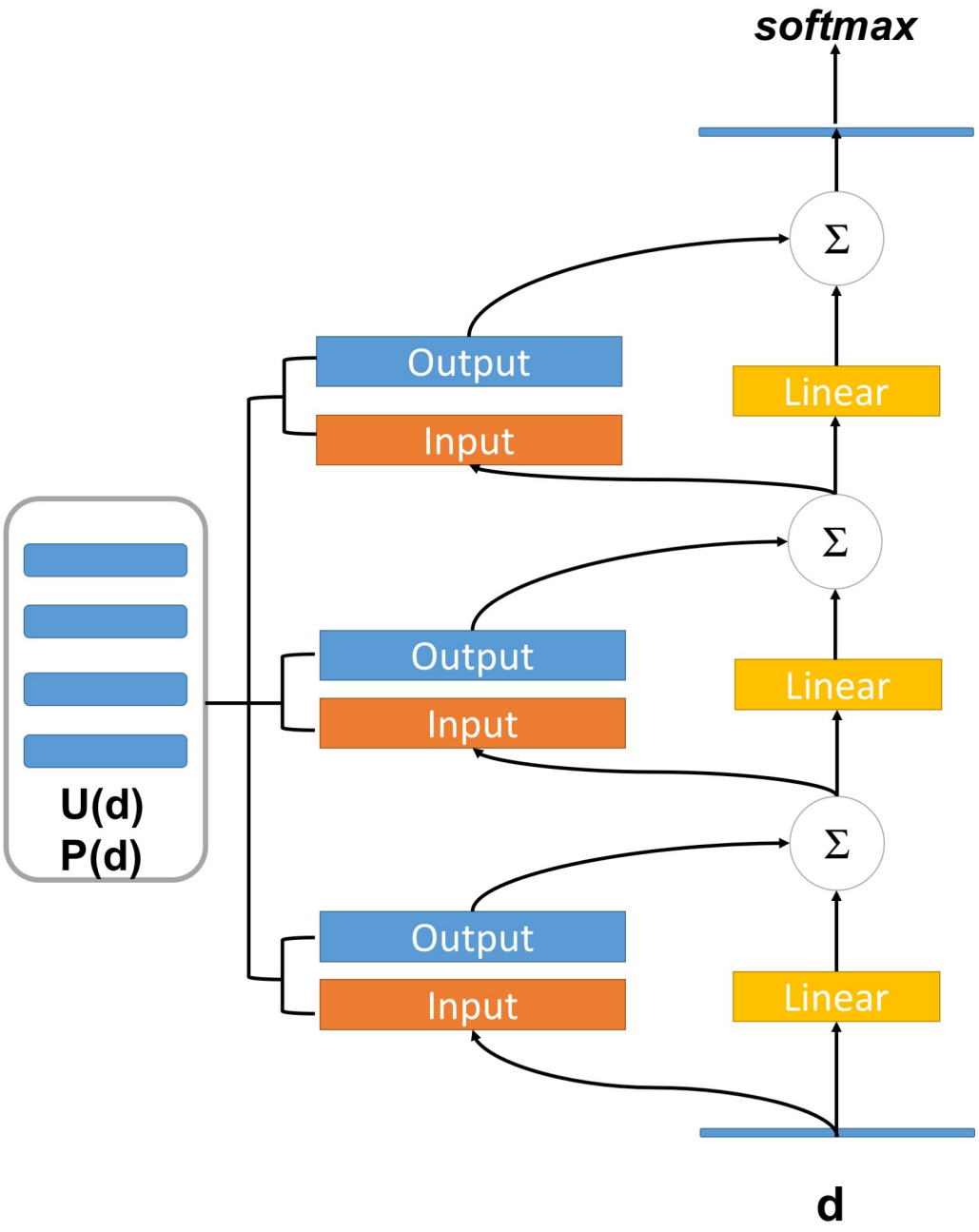


Figure 2.8: An illustration of User memory network (UPDMN)

written by a person may be subjective or biased towards his/her own preferences. Lenient users tend to give higher ratings than finicky ones even if they review the same products. Popular products do receive higher ratings than those unpopular ones because the aggregation of user reviews still shows the difference in opinion about different products. While users and products, both play crucial roles in sentiment analysis, they are fundamentally different. To explore different parts of profile information is becoming a heated research topic in personalized emotion analysis.

2.4 Chapter summary

In this chapter, the related background of this thesis is introduced. The related background, including methods for emotion and sentiment analysis, user profile construction, and incorporate user profile as subjectivity into consideration for emotion analysis. In terms of the use of personal profile related data in emotion analysis, there are three major problems in current research: (1) Current emotion analysis works have insufficient methods to incorporate appropriate linguistic features into emotion analysis model in social media and review text; (2) The current user profile construction methods mostly focus on learning from observed data instead of learning from both observed data and missing data to solve data sparseness issue; and (3) Current emotion analysis models handle user profile and product information in a unified model which may not be able to learn salient features of users and products effectively.

Chapter 3

Linguistically driven model for emotion analysis

Emotion analysis has been studied using different NLP techniques from a variety of linguistic perspectives such as semantic, syntactic, and cognitive properties [12, 9, 131, 243, 99, 157]. Past works primarily rely on morphology, syntactic or semantic orientation features to improve the task of emotion classification. But, the traditional feature engineering methods mostly work on text written in formal style [1]. Emotions can be encoded differently in different genre of text. For example, news is mostly written in formal style whereas social media text is written in causal style. The dramatic increase of SR text has led to a greater demand for emotion analysis for text of casual style. Typical social media text include micro-blog, tweets and barrage (real-time comments appearing on the screen with the flow of videos). Social media text are typically rich in emotion because they are often written as unfiltered immediate reactions to breaking events or expression of personal feelings. This chapter explores how to use features more suited for emotion analysis for this kind of text, especially for Chinese text.

Many traditional emotion analysis models extract text features from morpho-syntactic and semantic perspectives [70, 72], such as word-formation and syntactic constructions. However, social media text, especially Chinese text, has some characteristics which make emotion analysis more challenging. Firstly, social media text can be very short. This is

particularly true for Chinese as written text can consist of only a few characters or very short sentences [51]. Hence, only limited content and context information are available and thus there are insufficient cues for linguistic-driven models to learn. Secondly, social media text are normally written spontaneously as reaction to certain events and discussion issues. In a society which demand for more conservative manners in public, Chinese social media text are full of idiosyncrasies. This makes extracting semantic features even harder. Thirdly, neologism in social media increases the difficulty for extracting morpho-syntactic features. In particular, the newly coined phrases tend to contain different types of symbols, notation and scripts. Chinese social media text also tend to have symbols, words or phrases, from foreign languages as well as romanized notations, shorthands and Chinese in Pinyin forms, generally referred to as code-switch.

Despite the challenges, Chinese social media text are rich in emotion related orthographic features. Firstly, due to the relatively conservative social manner social media users tend to use Pinyin or English translations to express certain idea which may be sociologically or politically sensitive to avoid getting attention or deletion. For example, 民主(*minzhu*, democracy) is often represented by *minzu* or other similar misspelled pinyin sequences. Similarly, 政府(*zhenfu* 'government') may occur in the form of *zf* or *gov*. This phenomenon is generally regarded as a special type of code-switch [1]. Secondly, because punctuations and emoticons in Chinese social media are less censored [84], users are very creative in using them to express their feelings. For example: the ellipsis symbol "..." in formal text is used to indicate omission. However, in social media environment, it is generally used to show discontent. Also, "?!?!", as an unconventional use in formal text, is used to express surprise in social media text. Such characteristics in Chinese social media become a way of expressing emotions as a new writing style.

[1] Generally, we refer to code switch as the mixing of languages within a single document, the change between Chinese characters and Chinese Pinyin is called as writing system change. In order to keep the term simple, we use code switch to refer both phenomena.

To better handle social media text, this chapter approaches emotion analysis from a novel perspective by incorporating features associated with orthography including the consideration of shifts of symbols between language scripts and the use of stylistic variations such as unconventional use of punctuation marks. Firstly, based on observation from social media text, we propose two sets of features: orthographic features and morpho-syntactic features. These selected features are incorporated into a Support Vector Machine (SVM) to form our linguistic driven model. The model is evaluated by three different types of datasets to test the hypotheses as a cross-domain dataset comparison. The datasets include two informal social media text: one is the micro-blog dataset and the other is quite a different Bilibili dataset. The third dataset is the HIT news dataset [247]. Results show that orthographic features are indeed linked to emotion classification in social media text although they are not relevant to formal text. On the other hand, morpho-syntactic features contribute to emotion classification in formal style text. But, they contribute much less to social media text.

The rest of this chapter will the be organized as follows. Section 3.1 briefly summarizes related work. Section 3.2 describes our linguistic driven model. The evaluation of our proposed model is detailed in section 3.3. Section 3.4 concludes the chapters and provides future directions.

3.1 Related work

The related work in this chapter mainly include using linguistic features relevant to social media and review text for emotion analysis. Emotion analysis is generally approached based on semantic information because they offer assessment of the emotion value of a phrase for automatic classification [227]. Semantic orientation has been employed to estimate the positive or negative orientation of a phrase based on its association with positive or negative evaluations [75, 227]. Thus past studies focus on creating the lexicon of po-

larity adjectives and then the adjectives can be used to predict emotion polarity [75]. The task of determining subjectivity in reviews or comments thus also relies on the emotion polarity value of lexicon [76, 242, 241]. Overall, the semantic approach works well in the classification of reviews which have intact paragraphs such as movie reviews [223, 227].

It has been proved that semantics is closely associated with emotion analysis[97]. Socher et al.[202] proposed a matrix-vector recursive neural network model based on semantics compositionality. Lazaridou et al. [112] also proposed compositional distributional semantic models with morphemes as basic units. From the semantic perspective, sentiment analyzers in deep learning techniques have been improved [141, 51]. In machine learning models, the semantic perspective is frequently adopted in both supervised approaches [176, 15, 146] and unsupervised approaches [151, 125]. To achieve a better performance in classifying sentiments, supervised approaches have received more attention [166, 175]. Different linguistic levels of features have been implemented to improve performance such as bag-of-words features [50], semantic features [9, 131], syntactic properties [146, 167], and even cognitive features based on eye-movement patterns [99, 157].

Orthographic features like code-switch have been noted in emotion analysis as well [144, 230, 231]. Previous tasks, however, focus on strict definitions of code-switch, which refer to bilingual or multilingual switches. Other types of Code-switching have also been recognized to be relevant to emotions, particularly in Chinese social media [115, 237].

However, it should be noted that the above methods are implemented in formal style text in order to obtain sufficient linguistic information to improve performance. The current challenge is to deal with social media datasets that have very limited contextual information. Due to the pervasive spread of on line social networks, the interests on the task of emotion analysis for social media text has been increased [60, 12, 243]. Emotion classification for social media text has been proven to be challenging because most social media text has only limited contextual information [51]. As noted in Mishra et al. [157],

in social media text, subtleties at lexical, semantic, pragmatic, and syntactic level all have the possibility of influencing sentiments. Then the important question is which linguistic levels would have major factors, which will be discussed in the following sections.

3.2 Our proposed model

We include two sets of features for comparison. One set is orthographic features which targets the switches of symbols between different language systems; the other set captures morpho-syntactic properties.

3.2.1 Orthographic features

The one challenge in this task is that social media text frequently contain newly coined phrases. Different from formal and well-structured paragraphs in newspapers, datasets from social media posts such as micro-blog and particularly Bilibili contain rich code-switch and non-traditional uses of punctuations. These unexpected switches of symbols between writing systems in datasets are closely associated with expressions of particular emotions.

The features based on switches of symbols are detailed below. Each feature is represented by an abbreviated form for subsequent discussion.

- Switches of symbols between language scripts orthographically (Symbol Switches): including alternation between English and Chinese as in the case of code-switch, alternation between Pinyin abbreviation and characters, and alternation between Chinese characters and Pinyin, as shown in Table 3.1.

- Punctuation (?-mark for question marks,!-mark for exclamation marks, S-mark for suspension marks): When they are used to express strong emotion, they tend not to obey the standard rules in formal writing, which do not allow repetitions and com-

Type	Example
English-CN	没bird我(ignore me); 我好sad (I'm very sad); 我是不care (I don't care)
Character-abbreviation	我tnd翻了几千层评论 (I fucking went through thousands of comments); d丝(nerds/losers); nc粉(fans who admiring idols)
Char-Pinyin	Dalao们没人用这个(No experts will use this); 大家都是纯（hen）洁（wu）的 (Everyone is innocent (polluted)); 总有那么几个nao can (There are always some brain-damaged people.)

Table 3.1: Code-switch examples in Bilibili barrage

bined use. For example, '啊。。。'(ah...), '哇！！！！露的人力！！！！'(wow!!!!what human power), and '当然我是女的????'(of course I am a woman????).

- Emoji (E-mark): These are emoji characters as well as numeral numbers and other punctuation markers. For example, the combinations, *:3* and *23333*, are used for expressing smiles or laughs to reveal joy.

- Sentence length (SenLen): The length of a sentence is determined by the occurrence of punctuation markers. Since short text involve non-traditional use of punctuations, sentence length becomes a relevant indicator.

3.2.2 Morpho-syntactic features

Morpho-syntactic features have been proven to be effective in classifying event types in Chinese [34]. Morpho-syntactic features capture the grammatical properties of text. The morpho-syntactic features used in our experiments are listed below.

- Passive constructions (Pass): the occurrence of passive markers such as 被*bei* and

给*gei*.

- Disposal constructions (Dis): profiling a patient object and can be detected by markers such as 以*yi* and 把*ba*.
- Aspectual markers (Asp): defining the status of an event such as in progress and being completed.
- Double-object construction (DO): A verb takes both a direct and an indirect object.
- Relative clauses (RC): indicated by a relative clause marker.
- Negation (Neg)
- Postpositions (Post): The verb may take a post-position phrase.
- Prepositions (Prep): The verb may occur with a pre-position phrase. The indicators are the prepositions.
- Numeral phrases (Num): Quantity is specified by numeral-classifier phrases.
- Locative phrases (Loc): Location of an event is specified.
- V-V compounding (VV): The predicate of a main clause is two juxtaposed verbs.
- Transitivity (Vt): the transitivity of the verb.
- Word order (VO): The verb and object occur in either VO or OV word order.
- Chinese high-frequency markers (*yong*, *dui*, *gei*): including the instrumental marker 用*yong*, the goal marker 对*dui*, and the applicative marker 给*gei*.

The two sets of features target different linguistic aspects of a language in its actual use. The orthographic features reflect the stylistics and their correspondent social-pragmatic factors in the text, whereas the morpho-syntactic features emphasize on the grammatical

structures of the text. The salient distinction of the two sets of features is helpful to identify the most crucial linguistic level in emotion classification for different types of genres. It should be noted that the feature of word order in the morpho-syntactic set might be involved in code-switch when two languages employ different types of word order. Since both of Chinese and English have SVO as the primary word order, the two features, word order and code-switch, do not have dependent issues. Overall, the two sets of features do not have dependent issues; instead, they have independent linguistic motivations.

3.2.3 Feature extraction

Since two of our datasets contain social media text with short sentence, sentence fragment, and causal style representation, extracting the features would be difficult. We use the following methods to extract our proposed features.

- Direct extraction based on the specific symbols: This method can be applied to all punctuations, Emoji (E-mark), relative clauses, and Chinese high-frequency markers.

- Extracting based on manually designed rules: We manually define feature template to extract the features such as negation, postpositions, prepositions, location phrases, numerical phrases, and aspectual markers because so far no accurate parsers which are widely used available in Chinese.

- Extracting based on NLP processing tools: We use HIT-cloud[2] to perform part-of-speech tagging, syntactic parsing, dependency parsing, and named entity recognition. The features which are extracted by HIT-cloud tools include passive construction, double object construction, V-V compounding, transitivity, and word order.

[2] https://www.ltp-cloud.com/demo/

3.2.4 Emotion classification

The task of detecting emotion labels is modeled as a classification problem. In all three datasets, the emotion analysis is regarded as a multi-class classification problem based on emotion label schema of datasets. Because all three datasets are relative small in size, we choose SVM with linear kernel as our classifier for baseline model and use LibSVM [29] as the SVM implementation tool in order to test the effectiveness of the three datasets in emotion analysis. Specifically, we use hinge loss function, l2 penalty with linear kernel.

Classification tasks are conducted on Bilibili dataset (five categories: ”happy”,”sad”,”anger”,”fear” and ”surprise”), micro-blog dataset (five categories:”happy”, ”sad”, ”anger”, ”fear”, and ”surprise”.), and HIT news dataset (eight categories: “touched”, “empathy”, “boredom”, “anger”, “amusement”, “sadness”, “surprise”, and “warmness”). Precision (P), Recall (R) and Micro F-score (F) are included to measure the performance. 5-fold cross validation is used for all the three measures.

3.3 Experiments

We first describe three different datasets for evaluating our proposed model. Then we introduce the selected baseline models.

3.3.1 Datasets: from social media text to formal news text

Our experiments include three types of datasets, which range from social media text to news text. The style is range from causal style to formal style. Regarding the social media text, the dataset has been collected from Bilibili [3], one of the most popular websites in China for the youngster to share their video clips. Bilibili is famous for its commenting function because the platform allows a very large audience and creates a highly interactive platform for comments. The type of text are termed as barrage text as defined in the intro-

[3] https://www.bilibili.com/

duction. Barrage text contain fragments or incomplete phrases due to the real-time quick responding style in exchanges of information. Bilibili is also famous for neologisms. The newly coined phrases from Bilibili spread quickly to colloquial conversations. To structure the dataset, we crawled the comments from the top 17 video channels, ranked as the most popular channels by Bilibili on its official page. The comments were annotated by 5 annotators, who are native speakers of Chinese and have at least 5 years of experiences in using Bilibili. Their ages range from 18 to 22 years old so they are familiar with the newly coined phrases on this platform. The emotion labels are annotated in five categories, ”happy”,”sad”,”anger”,”fear” and ”surprise”. The dataset contains 10,482 annotated comments. In the 10,482 comments, 1,286 comments contain switches of symbols among different language systems, taking up 12.27 % of all comments. Regarding punctuations, 492 comments (4.65%) have question marks, 1,893 (17.89%) have exclamation marks. 358 comments have suspension points (3.38%), and 239 comments have emoji such as *:3* and number presentations *2333*, which take 2.20% of all comments.

For evaluation and comparison, we also used annotated formal style news dataset, named HIT news dataset(Harbin Institute of Technology news dataset) provided by Xu et al. [247] in our experiments. The HIT news dataset contains 23,385 pieces of news, with 35,2141 sentences tagged by the following eight categories, “touched”, “empathy”, “boredom”, “anger”, “amusement”, “sadness”, “surprise”, and “warmness”.

Between the social media text and the news text, we also include the dataset used in [237], which have the average sentence length in the middle and the style is also between this two sets. The dataset is retrieved from micro-blog, which is one of the famous SNS (Social Network Service) websites in China. The dataset include 4,195 code-switch posts for emotion annotation. The code-switch are identified by employing encoding code for each character in the post. For ease of reference, this dataset is termed as Mirco-blog. The dataset have five categories: ”happy”, ”sad”, ”anger”, ”fear”, and ”surprise”. This dataset is also taken from social media, but the have longer sentence length.

Length	Bibilibili	Weibo	HIT
Chinese	38.94	42.76	396.88
English	8.2	3.71	1.18

Table 3.2: The average sentence length of the three datasets

This study also aims to explore the association between the contribution of types of feature sets and types of datasets varying in text genre and sentence length. We thus include the three datasets, of which the average length in Chinese and English is provided in 3.2. In English, each unit is defined by the break of space. In Chinese, each unit is defined by the phrase parser [32]. As a social network dataset, Bilibili contains the most English components and has a typical casual style, whereas HIT contains the least English components and has a formal text style. In terms of the length in Chinese, HIT news has the longest Chinese sentences, while Bilibili has the shortest length. Mirco-blog is between the two but closer to the end of Bilibili.

3.3.2 Baseline models

The baseline model only uses the raw text features to process the text information in the baseline experiment. Regarding raw text features for the baseline model, we employ the widely used word weighing scheme in text mining problems, known as Term Frequency-times inverse document frequency (tf-idf). First, we segment each text file into words (for English splitting by space), and then we count times for each word that occurs in each document. Afterwards, we assign each word an integer id. Each unique word in our dictionary corresponds to a feature (descriptive feature) that we can further reduce the weight of more common words such as *the, is, an*, which occur in all document by using tf-idf. The proposed morpho-syntactic and orthographic features in section 3.2 are included in this model as additional features. We further compare with three different feature group combinations: the model using our proposed orthographic features only, the model using morpho-syntactic features only, and the model using both orthographic and

morpho-syntactic features.

Data set	Bilibili (social media)			Micro-blog (social media)			HIT news(formal)		
	F	P	R	F	P	R	F	P	R
SVM CS	*0.684*	*0.703*	*0.677*	*0.611*	*0.618*	*0.602*	*0.686*	*0.703*	*0.663*
Orthographic features (O)									
CS	0.710	0.730	0.702	0.632	0.639	0.625	0.683	0.703	0.663
...	0.701	0.719	0.693	0.629	0.632	0.625	0.682	0.703	0.661
Emoji	0.690	0.715	0.681	0.631	0.640	0.623	0.684	0.702	0.667
SenLen	0.701	0.718	0.694	0.629	0.637	0.622	0.681	0.700	0.663
Morpho-syntactic features (Ms)									
VO	0.707	0.739	0.698	**0.639**	**0.648**	**0.630**	0.695	0.712	0.680
Dis	0.664	0.688	0.656	0.620	0.625	0.615	0.687	0.711	0.665
Asp	0.673	0.692	0.666	0.624	0.628	0.620	0.689	0.712	0.667
DO	0.651	0.673	0.646	0.626	0.634	0.619	0.685	0.711	0.661
Feature combination (C)									
O + Ms	0.699	0.724	0.692	0.637	0.642	0.632	0.704	0.727	0.682
(p-value)	10^{-7}	10^{-7}	10^{-6}	10^{-7}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-8}
All Ms	0.692	0.709	0.680	0.630	0.638	0.622	**0.710**	**0.734**	**0.690**
(p-value)	0.001	0.001	0.010	10^{-6}	10^{-6}	10^{-7}	10^{-6}	10^{-7}	10^{-5}
All O	**0.722**	**0.739**	**0.707**	0.638	0.644	0.632	0.687	0.704	0.670
(p-value)	10^{-9}	10^{-8}	10^{-6}	10^{-8}	10^{-8}	10^{-9}	0.145	0.164	0.241

Table 3.3: Performance of emotion/sentiment analysis tasks on the three datasets

3.4 Results and discussion

The experimental results of the two sets of features are summarized at the bottom of Table 3.3. From the two sets of features, we select some representative features from each group for an in-depth discussion, as shown in Table 3.3. The two sets of features result in different performances in the three datasets. The set of orthographic-switch features contribute most to Bilibili dataset and then micro-blog dataset, but not much to HIT news. The results show that the features regarding switch-symbols enhance the emotion classification in varying degrees. The types of emotions are relevant to the changes of orthography particularly in social media text, but not much in longer formal style text. Among this set of features, the most effective feature for the Bilibili dataset study is Symbol-switch. In the

type of social media text, the linguistic cues are very limited since the text often contains sentence fragments. In this case, the orthographic switches are helpful indicators for specifying types of emotions. On the other hand, Symbol-switch is not the most relevant level for formal text. In terms of punctuation markers, they are helpful in emotion classification in Bilibili and Micro-blog as shown in the example of suspension markers, but they do not improve the task in HIT news. The results indicate that the use of punctuation is less restricted in social media text. Thus the creative combinations of punctuation markers, such as *!!!!!!*, *!!???*, and ..... are frequently used to express particular types of emotions. Regarding sentence length, it is an effective feature for sentiment classification in shorter text such as in Bilibili and micro-blogs, but it does not improve the emotion classification in HIT news. The contrast shows that the length of sentences is more flexible in social media text. The flexibility allows the users to use the length to mark their emotions as in the case of using sentence fragments to indicate the burst of emotions. However, the HIT news dataset allows no sentence fragments, so sentence length is not a helpful indicator of types of sentiments. Similarly, the emoji feature contributes to Bilibili and micro-blog, but not to HIT news. It indicates that the flexible combinations of symbols from different writing systems in the shorter text are associated with sentiments. Overall, the orthographic features effectively improve the performance of emotion classification in social media text. In the set of morpho-syntactic features, some features work well for all the three datasets, but most features only make improvement of emotion classification in longer text. Among the features, VO is the most effective one in improving the performance. It shows that the variation of VO is connected to types of sentiments and it is restricted neither by types of genres nor by the length of sentences. Nevertheless, most of the morpho-syntactic features do not improve the emotion classification in social media text although they work well for long text. As shown in 3.3, the disposal markers (Dis) do not improve the performance in Bilibili, but they contribute to the model of HIT and Micro-blog. Similarly, the aspectual markers (Asp) improve the performance in HIT news,

but they make the performance worse in Bilibili. In particular, the double-object construction (DO) largely decreases the performance in Bilibili. When all the morpho-syntactic features are incorporated, they help to improve the performance of all the three datasets. It should be noted that formal text such as HIT has more improvements. It is because formal text contain more morpho-syntactic cues, which can provide more information for the classification task.

When both orthographic features and morpho-syntactic features are incorporated, the incorporation improves the performance in the three datasets. Each set contributes to the improvement differently. In Bilibili, the orthographic features make the most improvement, but the morpho-syntactic features can only slightly improve the performance. In HIT news, the set of morpho-syntactic features effectively improve the performance, whereas the orthographic set only slightly contributes to the model. Regarding micro-blog, which has style and sentence length between Bilibili and HIT news, the contribution from the two sets of features is relatively balanced. The cross-domain comparison shows that the average sentence length indicates the richness of the linguistic cues from orthographic and morpho-syntactic aspects. As shown in our results, emotion classification in different genres is linked to different linguistic aspects. The orthographic features, which are socio-pragmatically motivated, contribute to enhance the performance in social media text; the morpho-syntactic features, which are structurally motivated, are more effective in formal style text.

The salient contrast between the two sets of features in the three different datasets is also relevant to neologism. Spontaneous social media text can accommodate a wide variety of newly coined phrases in a specific cultural setting, while newly coined phrases are less expected in long text. In shorter text as in Bilibili and micro-blog, neologism is relatively popular. The newly coined phrases tend to occur in a mixture of symbols from different language scripts such as nc粉*nc fen* ('fans who deeply admire their idols'), 激动ing *jidong ing* ('excited and up主*up zhu* ('the person who uploaded the video'). Due

to the characteristics of the short text, the orthographic features can help to solve the challenge from neologism generated in the Bilibili dataset.

3.5 Chapter summary

This chapter explores linguistic features associated with orthography for social media text. Our proposed model is based on the fact that different types of linguistic features works in emotion analysis tasks of different genre of text. We evaluate the orthographic features and morpho-syntactic features in three representative Chinese datasets. Results show that the orthographic features can effectively improve emotion in social media text, whereas the morpho-syntactic features play less important role. Evaluation on the formal text has opposite results. The morpho-syntactic features are the most important for formal text, while the orthographic features have limited contributions. As for the micro-blog dataset with the style between social media text and formal text, the two sets of features have relatively similar contributions to improve the performance. The performance in the three datasets shows that effectiveness can be achieved when features are designed to target the linguistic level responsible for emotion expressions. The proposed orthography features, motivated by linguistic theories, provide crucial information to reach a more precise emotion classification for social media text.

Chapter 4

Cognition grounded model for emotion analysis

Cognitive studies have indicated that not all words are contribute equally in the semantic and affective meaning of sentence [46]. Some words are more important than others in conveying semantic meanings. Attention models are proposed based on this premise to give different weights to different words in text.

Previous attention models for emotion analysis are built from information embedded in text including users, products and text in local context [214, 252, 33, 67]. However, many attention models are built by using local text features through distributional similarity which lack theoretical foundation.

The key in emotion analysis lies in its cognitive basis. Two phenomena rally behind the cognitive theories of emotion analysis [189]. First, people react to the same event with a variety of different emotions. The reaction is subjected to individuals' biases based on their cognitive experiences. Second, different events may trigger the same emotion as there are only a number of emotional reactions cognitively. Based on these two phenomena, we envision that cognition grounded data obtained in text reading should be helpful in building an attention model. Since attention models can be incorporated to build better sentiment analysis models, we can establish the indirect link between cognition grounded data and sentiment analysis.

In this chapter, we propose a novel cognition grounded attention (CGA) model for emotion analysis learned from cognition grounded eye-tracking data. Eye-tracking is the process of measuring either the point of gaze or the motion of an eye relative to the head[1]. In psycho-linguistics experiments, Barrett et al. [13] shows that readers are less likely to fixate on close-class words that are predictable from context. Readers also fixate longer on words which play significant semantic roles [47] in addition to infrequent words, ambiguous words, and morphological complex words [187]. Since reading time can be learned from an eye-tracking dataset, predicted reading time of words in its context can be used as indicators of attention weights.

We first build a regression model to map syntax, semantics, and context features of a word to its reading time based on eye-tracking data. We then apply the model to emotion analysis text to obtain the estimated reading time of words at the sentence level. The estimated reading time can then be used as the attention weights in its context to build the attention layer in a neural network based emotion analysis model. Evaluation on the four review based emotion analysis benchmark datasets (IMDB, Yelp 13, Yelp 14 and IMDB2[2]) show that our proposed model can significantly improve the performance compared to the state-of-the-art attention methods.

4.1 Related work

Eye-tracking has already been used in other research areas like face emotional recognition in computer vision [54]， aging study in developmental psychology [90]. In this section, we introduce two parts of tasks which are closely related to our cognition grounded attention model. The first part focuses on the previous studies about using eye-tracking data in NLP studies. The second part focuses on reading time prediction models in eye-tracking

[1] https://en.wikipedia.org/wiki/Eye-tracking

[2] We only evaluate our model on English dataset in this chapter because there are no open-accessible Chinese eye-tracking dataset.

data.

4.1.1 Eye-tracking data in NLP

Although there is no previous work on using eye-tracking for emotion analysis model, there are some researches connecting eye-tracking with other NLP tasks. Joshi et al.[98] proposes a novel metric called Sentiment Annotation Complexity for measuring sentiment annotation complexity based on eye-tracking data. Joshi et al.[98] show that word-sense-disambiguation (WSD) can make use of simultaneous eye-tracking. Eye-tracking data are also used to measure the difficulty in translation annotation [156]. Another research [159] presents a cognitive study of sentiment detection from the perspective of artificial intelligent where readers are tested as "sentiment readers". Mishra et al. [159] proposes a model in sentiment analysis and sarcasm detection by using eye-tracking data as a feature in addition to text features using naive-bayes (NB) and support vector machine (SVM) classifiers. Mishra et.al [157, 155, 160] propose a multi-task deep neural framework for document level sentiment analysis to predict the overall sentiment expressed in a document. multiple tasks include the learning of human gaze behavior and auxiliary linguistic tasks like part-of-speech tagging, detecting syntactic properties of words, or finding sarcastic information in the document. However, this model needs gaze information to be available in the sentiment analysis dataset. Gathering information for large sentiment datasets is too labor expensive.

4.1.2 Reading time prediction in eye-tracking data

It has been proved that gaze patterns (include reading time) during reading are strongly influenced by the feature of a word in terms of syntax, semantic, and discourse [101]. To predict reading time using eye-tracking data, Tomanek et al. [222] propose a regression model using linguistic features related to syntax and semantics for calibration. This work investigate different forms of textual context and linguistic complexity classes relative to

syntax and semantics. Firstly, this paper describes an empirical study where we observed the annotators' reading behavior while annotating a corpus. Then this study focus on building cost models for predicting eye-tracking time. The cost model is essentially a linear model with four feature groups: characters (basic), words, complexity, and semantics.

Hahn et al. [100] proposes a novel approach to model both skipping and reading using unsupervised method which combines neural attention with auto-encoding trained on raw text using reinforcement learning. This study is one of the first model to explain skipping behavior with computational models. This study evaluate the proposed neural attention model on the Dundee eye-tracking corpus [101], showing that it accurately predicts skipping behavior and reading times, is competitive with surprisal, and captures known qualitative features of human reading.

4.2 Proposed method

The design principle of our method is to add a CGA (Cognition Grounded Attention) model into a neural-network based LSTM sentiment classifier, a classifier that gives the state-of-the-art performance in sentiment analysis [190].

Let D be a collection of documents. A document d_k, $d_k \in D$, has m sentences $S_1, S_2, ...S_j, ..., S_m$. A sentence S_j is formed by a sequence of words $S_j = w_1^j w_2^j ... w_{l_j}^j$, where l_j is the length of S_j. The features of a word $w_i \in D$ form a feature vector $\vec{v}_{w_i} = [{F_1}^{w_i}, {F_2}^{w_i} .... {F_n}^{w_i}]$ where n is the feature space size. The purpose of document level sentiment classification is to project a document d_k into the target space of L class labels. Similarly, at the sentence level, the purpose is to map a sentence S_j into the target class space.

To build the CGA model, we need to first build a reading time prediction model for words within each sentence. Reading time is predicted based on word features and text features calibrated by eye-tracking data. Note that reading time from an eye-tracking

dataset cannot be used directly because the text of any eye-tracking dataset is too small for sufficient coverage.

Consequently, our method has four tasks:

- To predict the reading time of words using eye-tracking data with $\vec{v}_{w_i}$ as features;
- To build attention models based on predicted reading time at sentence level and document level;
- To integrate the proposed attention model with other attention models;
- To add the attention models into a LSTM based sentiment/emotion classifier (depend on the label of given dataset).

4.2.1 Modeling of reading time

To learn the reading time of words in a sentence, our method is based on regression analysis using eye-tracking data as dependent variables and context information in $\vec{v}_{w \in S_j}$ as independent variables.

In the eye-tracking process, a number of different time measures are included such as *first fixation duration*, *gaze duration*, and *total reading time*. In this work, we only use *the total reading time*.

We use features extracted from the context of an eye-tracking corpus to train the regression model. We select features based on the works from Demberg [47] and Tomanek [222] to include word features such as word length and POS tags as well as context level syntax and semantic features such as the total number of dominated nodes in a dependency parsing three, the maximum dependency distance, semantic category etc.

The features we selected are in four groups:

- Morphology features: number of characters and words per annotation phrase; words in a phrase starting with capital letters; words in a phrase consisting of capital letter

words which only have alphanumeric characters, or words which have punctuation symbols.

- Character features: number of named entity words and percentage of named entity words in the annotation phrase.

- Complexity features: syntactic complexity: number of dominated nodes, POS, n-gram probability, maximum dependency distance; semantic complexity: inverse document frequency; ambiguity (number of senses); general linguistic complexity: Flesch-Kincaid Readability Score [3].

- Context features: named entities in word context (preceding or following current phrase); abbreviation in word.

Given a word w in a sentence S_j, $w \in S_j$, and its feature vector $\vec{v}_{w \in S_j} = [F_1^w, F_2^w, ..., F_n^w]$, the regression model on eye-tracking data is a mapping function g between reading time $t_{w \in S_j}$ and $\vec{v}_{w \in S_j}$ as defined as follows:

$$t_{w \in S_j} = g(\alpha_1 F_1^w + \alpha_2 F_2^w + ... + \alpha_n F_n^w + b), \tag{4.1}$$

where $t_{w \in S_j}$ is the predicted reading time for w, α_i is the weight of feature F_i^w, and b is a constant. Note that the set of $\alpha_i (i = 1...n)$ forms the weight vector $\vec{\alpha}_w$ for $t_{w \in S_j}$. When $\vec{v}_{w \in S_j}$ takes scalar values, g can be an identity function and thus this model becomes a typical linear regression model. When $t_{w \in S_j}$ takes discrete values, g can be a logistic function and this model becomes a typical logistic regression model.

We set g to be the identity function. A objective function then becomes:

$$\min_{\vec{\alpha}} \sum_{a_i \in \vec{\alpha}}^{n} ||t_{w \in S_j} - y_{w \in S_j}||_2^2 + \lambda R(\vec{\alpha}), \tag{4.2}$$

[3] https://readable.io/

where $y_{w \in S_j}$ is the true eye-tracking values of reading time, $R(\vec{\alpha})$ is the regularization of $\vec{\alpha}$, and λ is the regularization weight. When $\lambda = 0$, the model degrades to a linear regression function. In this work, we evaluate the use of both the linear regression model and the Ridge regression model.

4.2.2 Building the attention based model

Once we have the predicted reading time for each words used in sentences, the attention model can be built with two components. The first component works at the sentence level to give different words different emphasis in a sentence. The second component works at the document level to give different sentences different emphasis in a document.

For a sentence $S_j = w_1 w_2 ... w_i ... w_{l_j}$ with length l_j, each word w_i in S_j has a corresponding reading time t_{w_i}. Let t_{S_j} denote the total reading time of S_j. Then,

$$t_{S_j} = \sum_{i=1, w_i \in S_j}^{l_j} t_{w_i}. \tag{4.3}$$

For sentence level attention, the CGA (Cognition Grounded Attention) weight for w_i in S_j, denoted as $A_{S_j:w_i}$, can be defined as:

$$A_{S_j:w_i} = \frac{t_{w_i}}{t_{S_j}}. \tag{4.4}$$

This sentence level attention model defined above gives more weights to words that have longer reading time relative to the total reading time of the sentence.

Let a document d_k, $d_k \in D$, be formed by a set of sentences $S_j = w_1 w_2 ... w_i ... w_{l_j}$. Now the CGA weight for a sentence S_j in d_k is defined as:

$$A_{d_k:S_j} = \frac{t_{S_j}}{\sum_{i=1}^{m} t_{S_i}}. \tag{4.5}$$

This aggregated document level attention model gives more weights to the sentences that have longer reading time relative to the total reading time of the document. Let $\vec{A}_{d_k}$ denote the document level attention weight vector. The size of $\vec{A}_{d_k}$ should be m, the number of sentences in d_k.

Let $\vec{S}_j$ denote the embedding of S_j in N dimensional space, where $S_j \in d_k$. Then, the set of sentence representations for d_k (contain m sentences) should be a matrix of size $m \times N$, denoted by $\hat{S}_{d_k}$. After the inclusion of the attention model, $\hat{S}_{d_k}$ should be:

$$\hat{S}_{d_k} = \vec{A}_{d_k} \vec{S}_j^T. \tag{4.6}$$

Let $\vec{d}_k$ denote the document embedding of d_k. Since $\vec{d}_k$ is an N dimensional vector, $\vec{d}_k$ can now be defined by the adjusted attention model as:

$$(\vec{d}_k)_i = \sum_{j=1}^{m} (\hat{S}_{d_k})_{i,j}. \tag{4.7}$$

4.2.3 Incorporation of other attention models

Since document embedding representation allows combined use of multiple attention mechanisms, it is to our advantage to incorporate different attention mechanisms to help in capturing different aspects of attentions. Generally speaking, different attention mechanisms can be incorporated either serially or in parallel.

In principle, any number of attention models can be included. As an example to illustrate the capability of our proposed method, we choose one state-of-the-art local attention model (shorthanded as LA) as an example for inclusion. The model is a semantic-based local attention model proposed by Yang [252] and also used by Chen et al. [33]. For inclusion serially in the LSTM layer, the attention weight is formulated as follows:

$$A^s{}_{S_j:} w_i = LA_{S_j:w_i} * A_{S_j:w_i}, \tag{4.8}$$

where $LA_{s_j:w_i}$ is the sentence level attention model by the Yang et al.'s [252] local attention model.

To incorporate LA in parallel mode, the attention weight can be formulated by:

$$A^p{}_{S_j:}w_i = LA_{S_j:}w_i + A_{S_j:}w_i. \tag{4.9}$$

The same method can be applied similarly in the document level. Similar methods can be used at document level.

4.3 Experiments and analysis

Our proposed CGA for emotion classification is evaluated on five document sets: The first three datasets IMDB, Yelp 13, and Yelp 14 are review text including user/product information developed by Tang et al. [214]. Since these three sets of data contains user/product information in each review, Tang's [214] work also used user/product information when building attention models. The fourth dataset IMDB2 is a collection of text on movie reviewers without user/product information [140]. The last dataset was originally developed for fake news detection (labeled FND), where the detection is on whether a piece of news by a speaker is fake or not. We use it to see if eye-tracking data can help with other text classification tasks in addition to sentiment [235].

Table 4.1 list the statistics of the datasets including number of classes, number of documents, number of users, number of products, and the average length of sentence. Note that in the FND dataset, user refers to speaker. We split train/development/test set in the rate of 8:1:1. The best configuration of the development dataset is used in the test set to obtain the final result.

Two commonly used performance evaluation metrics are used. The first one is accuracy and the second one is rooted mean square error (RMSE). [6] Let GR_i be the golden

[6] Normally accuracy is a problematic measure in highly unbalanced datasets. But in IMDB, the largest class only takes less than 20% of all instances. The most imbalanced data are Yelp 13 whose largest class is 41% among 5 classes and second largest is about 30%. IMDB has a 50/50 split for 2-classes.

Data	#class	#doc	#user	#pro	#len*[4]
IMDB	10	84,919	1,310	1,635	24.56
Yelp14	5	231,163	4,818	4,194	17.25
Yelp13	5	78,966	1,631	1,631	17.37
IMDB2	2	50,000	N/A	N/A	20.10
FND	6	12,836	12,022[5]	N/A	24.97

Table 4.1: Statistics of three benchmark datasets

sentiment rating, PR_i be the predicted sentiment rating, and T be the number of documents where $GR_i = PR_i$. Accuracy is then defined by:

$$Accuracy = \frac{T}{N}, \tag{4.10}$$

and RMSE is defined by:

$$RMSE = \sqrt{\sum_{i=1}^{N}(GR_i - PR_i)^2 * \frac{1}{N}}. \tag{4.11}$$

Note that RMSE is only suitable for range based labels. Hence, in our paper, RMSE is used only in IMDB, Yelp13, and Yelp14 for evaluation.

We train the skip-gram word embedding [154] on each dataset separately to initialize the word vectors. All embedding sizes on the model are set to 200, which is the same as [214, 215, 33, 249].

Three sets of experiments are conducted. The first is on the selection of the regression model for reading time prediction. The second set of experiments compares our proposed CGA with another sentiment analysis method which use text only. The third set of experiments evaluates the effectiveness of combining different attention models.

4.3.1 Reading time prediction

Reading time prediction, using regression models, are trained from eye-tracking data. In this work, we use three sets of public available eye-tracking data. Ideally, an eye-tracking

	Sentences	Tokens	Participants
Mishra[158] (M)	994	68543	7
Dundee (D)	2,368	51,502	10
GECO (G)	4,934	774,015	17

Table 4.2: General statistics of three eye-tracking corpus

corpus built from on-line reviews is more suitable for our experiments. But, we can only work with what is available. Their lengths in terms sentence and tokens as well as the number of participants are listed in **Table 4.2**.

Through our regression models, we learn to predict reading time from lexical and context features as discussed in section 4.2.1. We take the first 90% of sentences as training data and the rest 10% as test data. We compare our regression model with more complex deep learning based regression models in each of the three eye-tracking datasets. [7]

In addition to the linear regression (LR) model and the Ridge regression (RR) model, we also choose the Support Vector Machine (SVM) model with linear kernel, the Recurrent Neural Network (RNN) model and the Long Short Term Memory (LSTM) model for regression learning. For both models, there are two versions. The basic version inputs the extracted feature sets as word representation, labeled as SVM-1, RNN-1 and LSTM-1, respectively. The second version takes word embedding (dimension set to 200) [179] as the initial word representation input, labeled as SVM-2, RNN-2 and LSTM-2, respectively. The configuration that performs the best for each model is selected and the performance results are listed in Table 4.3. Data in **Table 4.3** are in milliseconds.

Table 4.3 shows that RR gives the best result in all three datasets, and both regression models outperform SVM and deep learning based models. The reason that RR has the best performance in all the three datasets is that regularization in RR reduces the over-fitting problem. Results of SVM and deep learning model with word embedding initialization partly support the fact that reading time are more dependent on micro level syntax and

[7] Mishra et al. [159] only provides fixation time. Fixation time is used when training by this set of eye-tracking data.

	GECO	DUNDEE	Mishra
RR	69.47	70.52	84.22
LR	72.47	73.52	87.25
SVM-1	73.46	77.50	88.96
RNN-1	75.47	83.52	96.23
LSTM-1	79.47	84.52	114.25
SVM-2	78.47	82.52	87.92
RNN-2	79.57	86.47	101.25
LSTM-2	83.88	95.88	122.27

Table 4.3: RMSE for reading time prediction

semantic feature of a word, such as number of letters in word and complexity score of the word instead of deep level global context features.

We also use correlation coefficient (generally shorten as RR in statistics) to describe the relationship between predicted reading time and actual reading time in eye-tracking data [8]. In the three eye-tracking datasets, RR can achieve coefficient of determination[9] at 0.32, 0.30 and 0.27 in three eye-tracking datasets. The features, their types and the corresponding coefficients in RR are shown in **Table 4.4**. Again, the features shown in **Table 4.4** are microlevel features.

Feature_Name	Type	Cofficient
Number of letters	Num	22.441
Start with capital letter	Bool	1.910
Capital letters only	Bool	161.580
Have alphanumeric letters	Bool	6.020
Is punctuation	Bool	-8.930
Is abbreviation	Bool	10.551
Is entity-critical word	Bool	7.612
Number of dominated nodes	Num	0.980
Max dependency distance	Num	1.982
Inverse document frequency	Num	-9.291
Number of senses in wordnet	Num	7.494
Complexity score	Num	57.240
Constant	Num	239.910

Table 4.4: Major features used for RR on eye-tracking data.

10

[8] $https://en.wikipedia.org/wiki/Correlation_coefficient$

[9] https://en.wikipedia.org/wiki/Coefficient_of_determination

4.3.2 Comparison of different sentiment classification methods

		IMDB		Yelp13		Yelp14		IMDB2	FND
		ACC	RMSE	ACC	RMSE	ACC	RMSE	ACC	ACC
G 1	Majority	0.196	2.495	0.411	1.060	0.392	1.097	0.500	0.204
	Trigram	0.399	1.783	0.569	0.814	0.577	0.804	0.848	0.208
	TextFeature	0.402	1.793	0.556	0.845	0.572	0.801	0.841	0.227
	AveWord2vec	0.304	1.985	0.526	0.898	0.531	0.893	0.831	0.226
G 2	SSWE+SVM	0.312	1.973	0.549	0.849	0.557	0.851	0.853	0.231
	Doc2vec	0.314	1.814	0.554	0.832	0.564	0.802	0.863	0.225
	RNTN+RNN	0.401	1.764	0.574	0.804	0.582	0.821	0.869	0.241
	CLSTM	0.421	1.549	0.592	0.769	0.594	0.766	0.872	0.245
	B-CLSTM	0.462	1.453	0.619	0.705	0.592	0.741	0.878	0.247
	LSTM	0.443	1.465	0.627	0.701	0.637	0.686	0.870	0.241
G 3	LSTM+LA	0.487	1.381	0.631	0.706	0.631	0.715	0.885	0.255
CGAs LSTM+	CGA^{M}	0.447	1.495	0.610	0.746	0.613	0.768	0.868	0.255
	CGA^{D}	0.468	1.419	0.623	0.706	0.628	0.702	0.886	0.267
	CGA^{G}(W)	0.469	1.414	0.633	0.700	0.633	0.688	0.884	0.268
	CGA^{G}(S)	0.471	1.412	0.634	0.699	0.635	0.687	0.885	0.269
	CGA^{G}	**0.489**	**1.365**	**0.638**	**0.697**	**0.641**	**0.678**	**0.894**	**0.278**

Table 4.5: Evaluation on sentiment classification using only review text for training

Because the features used in our model are all text based, we compare CGA with three groups of baseline methods which also only use review text for learning. Group 1 methods include commonly known linguistic and context features for SVM classifiers. Group 2 includes recent sentiment classification algorithms which are top performers using review text for training, without attention mechanisms. Group 3 includes two state-of-the-art attention methods.

- **Majority** — A simple majority based classifier based on sentence labels.
- **Trigram** — An SVM classifier using unigrams/bigrams/trigram as features.
- **Text feature** — An SVM classifier using word level and context level features, such as n-gram and sentiment lexicons.

- **AvgWordvec** — An SVM classifier that takes the average of word embeddings in Word2Vec as document embedding.

Here is a list of Group 2 methods:

- **SSWE** [218] — An SVM classifier using sentiment specific word embedding.
- **RNTN+RNN** [203] — A Recursive Neural Tensor Network (RNTN) to represent sentences and trained with RNN model.
- **Paragraph vector(Doc2vec)** [114] — An SVM classifier using document embedding as features.
- **CLSTM** [245] — A Cached LSTM to capture the overall semantic information in long text. The two variations include regular **CLSTM** and bi-directional **B-CLSTM**.

Here is a list of Group 3 methods which use attention mechanism:

- **LSTM+LA** [33] — State-of-the-art LSTM using local context as attention mechanism in both sentence level and document level.
- **LSTM+UPA** [33] — A state-of-the-art LSTM including LA as well as user/product as attention mechanism at both sentence level and document level. This method only used when user/product information is available.

Our proposed CGA model has several variations as explained below.

- **LSTM+CGA** — An LSTM classifier using only CGA model at sentence level and document level. Based on the three eye-tracking datasets (GECO, DUNDEE and Mishra's) for reading time prediction, we label the same model by different training data as **LSTM+CGA**^{G},**LSTM+CGA**^{D} and **LSTM+CGA**^{M} (G,D,M represent three different eye-tracking datasets: GECO, DUNDEE and Mishra's). For

LSTM+CGA^G, we evaluate the importance of word level attention and sentence level attention by using attention mechanism only on word level (LSTM+CGA^G(W)) or sentence level (LSTM+CGA^G(S)).

- **LSTM+CGA+LA**^{G} — An LSTM based classifier using both the CGA model and Yang et al.'s [249] local text context based attention model (LA) [33]. Since combining methods can either be serial or in parallel, there are actually two corresponding variations: LSTM+CGA+LA$_s^G$ and LSTM+CGA+LA$_p^G$.

- **LSTM+CGA+UPA**^{G} — The same framework to **LSTM+CGA+LA**^{G} with an additional user/product attention. The user/production attention is built from user and product information for all datasets except IMDB2. The two corresponding variations are LSTM+CGA+UPA$_s^G$ and LSTM+CGA+UPA$_p^G$.

	IMDB		Yelp13		Yelp14		IMDB2	FND
	ACC	RMSE	ACC	RMSE	ACC	RMSE	ACC	ACC
LSTM+LA	0.487	1.381	0.631	0.706	0.631	0.715	0.885	0.255
LSTM+CGA^G	0.489	1.365	0.638	0.697	0.641	0.678	0.894	0.278
LSTM+CGA+LA$_s^G$	0.488	1.369	0.633	0.706	0.643	0.672	0.898	0.281
LSTM+CGA+LA$_p^G$	0.492	1.362	0.639	0.696	0.639	0.675	**0.901**	0.283
LSTM+UPA	**0.533**	1.281	0.650	0.692	0.667	0.654	N/A	0.289
LSTM+CGA+UPA$_s^G$	0.523	**1.277**	0.654	0.693	0.664	0.645	N/A	0.291
LSTM+CGA+UPA$_p^G$	0.521	1.278	**0.655**	**0.685**	**0.668**	**0.644**	N/A	**0.293**

Table 4.6: Evaluation on sentiment classification on using dual attention models

We split train/development/test set at the rate of 8:1:1 to keep the same as our comparison baselines. The best configuration of the development dataset is used in the test set to obtain the final result. **Table 4.5** shows the performance of the three groups using review text without user/product information. Among all the reference methods that do not use any attention mechanism including all methods in Group 1 and Group 2, LSTM is the best performer. This shows the advantage of using deep learning in recent development.

		IMDB		Yelp13		Yelp14		IMDB2	FND
Group	Lexicon	ACC	RMSE	ACC	RMSE	ACC	RMSE	ACC	ACC
LSTM	N/A	0.443	1.465	0.627	0.701	0.637	0.686	0.870	0.241
SA lexicon (Group 1)	VADER	0.481	1.371	0.631	0.705	0.624	0.697	0.883	0.259
	SWN	0.341	1.701	0.607	0.747	0.611	0.733	0.854	0.237
	SN	0.372	1.608	0.614	0.733	0.601	0.734	0.851	0.239
VAD methods (Group 2)	ANEW	0.467	1.362	0.626	0.704	0.626	0.699	0.890	0.260
	EPA	0.469	1.369	0.631	0.706	0.627	0.704	0.891	0.259
	DAL	0.471	1.376	0.626	0.702	0.631	0.684	0.884	0.258
Others (Group 3)	CON	0.458	1.435	0.635	0.684	0.625	0.694	0.886	0.264
	P meaning	0.460	0.374	0.630	0.687	0.624	0.701	0.877	0.257
Eye-track	Eye-track	**0.489**	**1.365**	**0.638**	**0.697**	**0.641**	**0.678**	**0.894**	**0.278**

Table 4.7: Comparison with other lexicons without using user/product information

LSTM+LA [33] in Group 3 is the state-of-the-art method which uses local attention mechanism to improve performance significantly compare to all methods in Group 1 and Group 2. Among our CGA based variations, using the GECO dataset gives the best result outperforming LSTM+LA in all three datasets. LSTM+CGA^G has significant improvement over LSTM+LA with p values of $p < 0.016$ on IMDB, $p < 0.0019$ on Yelp 13, $p < 0.00023$ on Yelp 14 ,and $p < 10^{-9}$ on FND. LSTM+CGA^G has the best result compared to the other two variations because GECO has larger participant size. Its text genre is also closer to the review datasets for sentiment analysis. For CGA model, word level attention and sentence level attention have a similar contribution to the improvement in all five datasets. Compared to LSTM model, the improvements bring by word level attention and sentence level attention are nearly the same. The results in table 4.5 proves that the additional cognition grounded data can boost the attention model to improve the performance of sentiment analysis.

In the third set of experiments, we compare our LSTM+CGA model with the combination of other attention models including the LA model and the UPA model as shown in **Table 4.6**. Since the GECO dataset gives the best performance as shown in previous experiments, results given in **Table 4.6** show the performance of LSTM+CGA using only

the GECO dataset. Note that UPA is an enhanced version of LA based on additional user/product information. So it works only if user/product information is available. Such data is provided in the first three datasets. For the FND dataset, speaker information is used to replace user information, and there is no product information.

Table 4.6 shows that among all three single attention models, UPA outperforms both LA and CGA in the first three datasets. This is easy to understand as UPA already included LA and it has additional information from users and products for its attention model.

The combined method of CGA with UPA can still further improve performance. When CGA+UPA is combined in parallel, it has the best performance for Yelp13, Yelp14, and FND (with p value of 0.027 ,0.032 and 0.0017 respectively compare to LSTM+UPA). In the IMDB dataset, however, UPA has the best performance. This may be because user/product information is more effective in the movie review IMDB dataset which is more subjective.

Since the UPA model works only if user/product information is available, for IMDB2, which does not have user/product information, only CGA and LA models work and the combined use of CGA+LA gives the best performance. Experiment indicates that incorporating in different aspects of attention is commendable. As the best result, the CGA model can work with others to take the full advantage of attention models in neural network based sentiment analysis.

4.3.3 Comparison of attention models based on other affective lexicons

Other lexicon-based resources can also serve as knowledge to build attention models. In [236], different lexicons were used to build attention models. Sentiment lexicons can be used directly to build attention modes for sentiment analysis by simply taking the sentiment values as attention weights. In other words, we can build LSMT+CGA with t_{w_i} replaced by sentiment values in a sentiment lexicon. In this experiment group, we com-

	IMDB		YELP13		YELP14		FND
	ACC	RMSE	ACC	RMSE	ACC	RMSE	ACC
UPA only	0.533	1.281	0.650	0.692	0.667	0.654	0.289
VADER	0.515	1.318	0.647	0.681	0.654	0.678	0.286
SWN 3	0.423	1.501	0.624	0.730	0.647	0.688	0.256
SN 4	0.433	1.487	0.620	0.743	0.648	0.667	0.258
ANEW	0.515	1.328	0.648	0.679	0.661	0.671	0.285
EPA	0.514	1.334	0.648	0.675	0.651	0.675	0.286
DAL	0.518	1.328	0.644	0.694	0.663	0.672	0.288
CON	0.518	1.303	0.647	0.681	0.661	0.671	0.285
P senses	0.515	1.308	0.645	0.683	0.659	0.670	0.283
Eye-track	**0.521**	**1.278**	**0.655**	**0.685**	**0.668**	**0.644**	**0.293**

Table 4.8: Compare with other attention mechanism in dual attention mechanism (with UAP+P)

	The	XXX	hotel	is	lucky	to	receive	2stars	from	me	considering	Pun
VADER	0.063	0.020	0.079	0.148	0.264	0.146	0.174	0.020	0.097	0.025	0.007	0.000
SWN 3	0.000	0.050	0.000	0.040	0.880	NA	0.040	0.010	0.000	0.010	0.020	0.000
SN 4	0.033	0.020	0.039	0.105	0.508	NA	0.061	0.010	0.099	0.133	0.022	0.000
ANEW	0.098	0.010	0.120	0.107	0.133	0.116	0.129	0.010	0.093	0.105	0.100	0.000
EPA	0.033	0.040	0.123	0.120	0.151	0.127	0.158	0.020	0.116	0.114	0.060	0.000
DAL	0.104	0.030	0.128	0.099	0.150	0.094	0.145	0.030	0.084	0.111	0.084	0.000
CON	0.155	0.030	0.204	0.066	0.073	0.064	0.111	0.040	0.076	0.179	0.072	0.000
P sense	0.105	0.040	0.103	0.104	0.109	0.077	0.107	0.030	0.122	0.138	0.136	0.000
Eye-track	0.070	0.086	0.078	0.082	0.078	0.072	0.088	0.116	0.071	0.078	0.082	0.082

Table 4.9: Case study on attention weights of using other lexicons and eye-tracking data

pare the use of eye-tracking data with other lexicons used by Li et al.[123]. We divided the lexicons into three groups.

The first group include commonly used sentiment oremotion lexicons:

- VADER [59] is sentiment lexicons annotated with intensity and VADER also contains standard deviation of the annotation process.
- SentiWordNet (shorten as SWN) [7] is a lexical resource for opinion mining. SentiWordNet assigns to each synset of WordNet [102] with three sentiment scores: positivity, negativity, and objectivity.
- SenticNet (shorten as SN)[27] provides a set of semantics, syntaxes, and polarity associated with 50,000 natural language concepts.

The second group includes three multi-dimensional affective lexicons. In these three affective lexicons, at least one dimension is directly link to sentiment. Thus, data in that dimension is used to serve as sentiment values.

- ANEW (the affective norms for English)[238] provides a set of normative emotional ratings for a large number of words in the English language. This set of verbal materials have been rated in terms of pleasure, arousal, and dominance to complement the existing International Affective Picture System. The extended version of ANEW lexicon consist of 13,915 words. ANEW is built based on the Valance-arousal-dominance schema (VAD) framework. The valence dimension can directly serve as sentiment.

- EPA (evaluation, potency, and activity) [77] is annotated in the three dimensions of evaluation, potency, and activity. In those three dimension, evaluation is close related to sentiment. Here the evaluation dimension is close to sentiment and it can be used to approximate sentiment.

- DAL (The Dictionary of Affect in Language) [87], a lexicon annotated in the dimensions of pleasantness-activation-imagery contains 8,742 terms. The Pleasant dimension can directly serve as the sentiment dimension.

The third group includes two sets of lexicons: one is to measure concreteness of concept terms, and the other is to measure perceptual sense which measures in cognition. They are evaluated to see how cognition linked lexicons can help in sentiment analysis.

- Concreteness (shorten as CON) [24] is annotated on the degree of concreteness or abstractness of a word through crowdsourcing.

- Perceptual sense (shorten as P sense) [137, 138] is annotated with perceptual strength of a target word by feeling through five sensations (touch, hearing, seeing, smelling ,and tasting).

Table 4.7 shows the comparison in the situation that user/product situation is not available. We can observed that nearly all sentiment lexicon except SentiWordnet and SentNet can outperform regular LSTM in four datasets. But all lexicons do not match the performance of LA and CGA models. In **Table 4.8**, we evaluate attention models based on these lexicons by perform dual attention mechanism with user/product attention. **Table 4.8** shows similar performance result which shows that using lexicon resources alone do not match up with LA based and CGA methods. The likely reason for this is that the sentiment values of each word in these lexicons are context-independent. That is, their values are fixed in the lexicon relative only to different entries in the same lexicons. On the other hand, the attention weight of each word in a sentence should be context-related. In other words, the attention weights of certainly words should be relative to other words in the same sentence (and/or documents) which is how they are produced in both LA based and CGA based methods. This is the main reason to explain the underperformance of lexicon based methods.

4.3.4 Case study

A random sentence sample *'The Shelton hotel is lucky to receive 2stars from me considering ...'* is taken from Yelp13 dataset to demonstrate the difference in the three attention mechanisms, i.e. local text (LA), cognition-based (CGA), and user/product attention(UPA). **Figure 4.1** shows visually the difference in attention weights of the three models.

The attention weights of words in the LA model does not change much. CGA, on the other hand, gives higher weights to the sentiment linked word *2stars* and the verb *receive*. These two words do play significant roles as an indirect object and a main verb, respectively. This case shows that CGA does a better job in capturing micro level information in the sentence level. This support the experimental results in **Table 4.5** and **Table 4.6**.

Table 4.9 compares our CGA with attention models based on other lexicons. We can

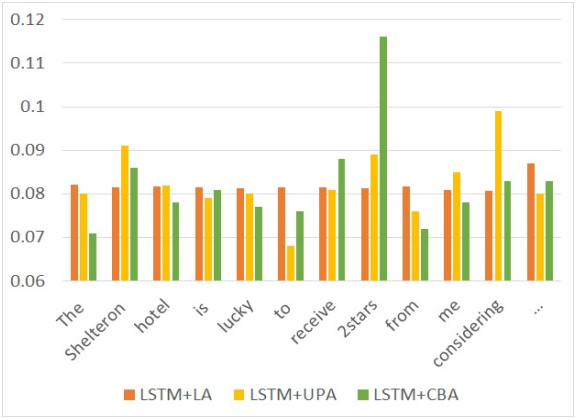


Figure 4.1: Case Study on attention weights in three different attention mechanisms

observe that Sentiwordnet and Sentinet give the sentiment word ”lucky” a very heavy weight while another words received relative low weight.This partly explains why these two lexicon achieves lowest performances in all lexicons. VADER, EPA and DAL give relative high weight to notional words. But they assign very low weight to functional words. This result indicates that the effect of function words should not be under estimated.

For group 3 lexicons concreteness and perceptual are not in sentiment space. But they still encode some valid semantic information as useful knowledge. For concreteness value, the subject of sentence ”hotel” receives the highest weight in all words. But in eye-tracking data and user/product attention, ”hotel” does not have a particular heavy weight.

4.4 Chapter summary

In this chapter, we propose a novel cognition grounded attention (CGA) model to improve the state-of-the-art neural emotion analysis model through cognition grounded eye-tracking data. A simple and effective regression model is used to predict reading time using both eye-tracking data and local text features. The predicted reading time is then used to build an attention layer in neural emotion analysis models. The CGA considers both reading time and other syntactic and context features. The CGA model also considers attention at both sentence level and document.

Evaluation on benchmark datasets validates the effectiveness of CGA method in sentiment analysis and related tasks. Our method clearly outperforms other state-of-the-art methods that use local context information to build their attention models. The CGA mechanism can also be combined with other attention mechanisms to provide room for further improvement. We compare the use of eye-tracking data with other lexical resources including emotion lexicons, dimension based affective lexicons, and other cognition based resources. One important reason that our CGA model prevails over other lexicon resources is that CGA can extract context relevant information including both sentence level context and document level context.

Our work also indicates that both the quality and the scale of eye-tracking data have great influence on the effectiveness of the CGA model. We anticipate even greater improvement with a larger scale eye-tracking data in similar genre as the emotion analysis text. This work validates the effectiveness of cognition grounded data in building attention models. More importantly, we bridge the gap between sentiment analysis and cognitive process by using cognition grounded data to build attention model and subsequently improve sentiment analysis models.

Chapter 5

User profile construction

User profile information can be represented in an explicit form or an implicit form. The archive page of a user in social media platforms is the most common form to explicitly specify a user profile in a structured way. Structured user profile includes factual information such as gender, age, education, spouse etc. Although explicit user profile can be readily used in many research and applications, such explicit information is sparsely available. Hence, researchers start to extract implicit user profiles from information obtained by recorded user activities contained in either semi-structured or unstructured data [20, 221]. User activities are mainly formed from two components. The first component is user generated text (shorten as user text) such as comments, posts, and status. These text often contain strong evidence about his preferences on events and entities. The second component is user established social network links (shorten as user links), which provides a clear social identity and relationship to others in the network.

As the result of learning from user activities, user profiles can be represented either as as discrete labels through information extraction [185] or as a dense vector using embedding methods [51, 225]. Since user profiles are increasingly complicated in social media, information extraction can only capture very limited information. Thus, representing user profiles by user embedding (representing user profile through dense vector), is becoming more popular. The representation of user embedding, can encode different types of pro-

file information without prior knowledge of their types. However, embedding methods, as deep learning methods, require a large amount of data during the training process[219].

However, data sparseness is extremely serious in user generated text on the Internet. In general, user activities follow long tail distributions [116], a few active users contribute to the majority of observed activities whereas the 'silent majority' only show very limited activities [63]. According to official statistics, there are 313 millions of twitter users in June 2016, but only 95.5 million users (about 30.3%) who have tweeted at least once in the first half of 2016, while the top 1% of frequent twitter users produce about 80% percent of tweets [1]. This long tail characteristics implies that learning user profiles using their explicit activities as observed data is not feasible. So, in this work, we try to address the data sparseness issue in two directions. The first direction is to make use of the missing-not-in-random assumption [83] to learn their profiles even if they are silent. The second direction is to make use of social network links to extend a user's context.

Using explicit data is based on the assumption that the probability of a piece of text to be missing is independent on the text itself [129]. However, most text are missing not at random [205]. User profile are not only encoded in observed text but also in missing text.

Evidences in both psychology and cognitive science study back the missing not at random hypothesis [83]. Stimulus generalization theory [177] believes that agents respond to things that are similar to the conditioned stimulus, and the way to respond is the same even if a different stimulus is received. Meanwhile, in cognitive science, the halo effect [168] shows that people do have cognitive bias in which an observer's impression of a person, a company or a team influences the observer's feeling of the target's characters. So, the stimulus generalization and halo effects imply that silence as a user behavior still encode user preferences. Furthermore, their inclination can be learned from the relationship between missing comments and observed text as well as from their relationship to other users.

[1] https://www.statista.com/statistics/234245/twitter-usage-frequency-in-the-united-states/

Extending user context through social network links is based on the homophily effect [149] that individuals who are linked on social networks tend to share common characteristics. Link information is considered as structured information links that connect nodes in a network naturally form a graph. In social networks, user link information has higher frequency. According to statistics, only 30.3% of users who have tweeted at least once in the first half of 2016, but on average each twitter user have 453 follower/follower connections [2]. Not only user links have higher frequency compare to user generated texts, user links can connect users with other users. If one user does not have enough text information to infer, we can expand it to the text of other connected users based on link information. Hence, learning user profile from both link and text information can also help to solve the data sparseness issue.

Based on these observations, we first explore a novel approach to predict user preferences by learning from both observed text and missing text based on the missing-not-at-random hypothesis. To further leverage on social network links available in social network, we explore methods to extend the context of user profiles through network links. We propose a novel approach to learn node embedding through a joint learning framework of both network links and text associated with nodes, the proposed models are designed for both homogeneous network and heterogeneous network.

The rest of this chapter is organized as follows: Section 5.1 gives related work. Section 5.2 presents our proposed work to learn user profile from unobserved data. Section 5.3 presents we proposed a model to extend the context of user profiles through network links. Section 5.4 is the conclusion.

[2] https://kickfactory.com/blog/average-twitter-followers-updated-2016/

5.1 Related work

The related work of this chapter mainly include two parts, the first part is learning user profile from both observed text and missing text. The second part is learning user profile from both network links and text associated with nodes.

5.1.1 Learning from both observed text and missing text

The general missing data problem or the long tail problem has been studied mostly in information retrieval and recommendation systems. Hu [83] proposes a model treating the data as an indication of positive and negative preference associated with rating confidence level, and then preform weighted matrix factorization (WMF) to get both user and item representations, WMF gives a smoothing weight to missing data. Stack [205] shows that the absence of rating carries useful information for improving the top-k rate of all items. Ma [139] integrates user social networks into a joint probabilistic matrix factorization model to solve data sparsity. In general NLP tasks, the missing data problem was first highlighted by Dagan [42] to estimate the probability of co-occurrences that do not occur in the training data. Guo [68] proposes a similar weighted textual matrix factorization model (WTMF) into sentence similarity tasks and incorporates external knowledge base for similarity measures to simulate different levels of missing data [69]. However, for resource poor languages like Chinese or in social media which is written in informal style, external knowledge may not be readily available.

5.1.2 Learning from both network links and text

In contrast to homogeneous networks, a heterogeneous network has more than one type of nodes such as users and videos. Different types of information can also be associated with different types of nodes such as text, attributes, and multi-media contents. An effective approach is to embed all types of nodes in a network as low dimensional vec-

tors. Thus, when using node embedding as node representations, downstream tasks such as information retrieval, recommendation and node classification, etc., can be conducted in fixed dimensional space [232]. Long et al.[133] combine user and text information in the Hupu network for user preference identification. Complex methods, such as the Community-enhanced Network Representation (CENE) [225], leverage both network link information and text information by modeling text as a special kind of nodes to optimize the probabilities of heterogeneous links. Tu et al. [224] propose a state-of-the-art Context Aware Network Embedding (CANE) model to extract context information with an attention mechanism for text embedding. But CANE was proposed for a homogeneous network. For heterogeneous networks having multiple types of nodes, Gui et al. [67] used a large-scale network embedding model initially proposed by Tang et al.[219] to explore user and product representations. However, when text information is included, comments written by the same user at different times, or comments made by different users of the same product node are treated as isolated text units non-indiscriminately. Even though individual comments can be short, a collection of them, as a document set to each node, can give more comprehensive information of the node. There are yet methods to explore the use of document information in text embedding for the learning of network embedding. Chang et al.[30] demonstrate that both link information and other rich content of text, and other information in a heterogeneous network can be captured by deep neural network approach, and a deep neural network is applied to represent heterogeneous networks which contain both text and picture information.

5.2 Learning user profile from observed text and missing text

Inspired by the stimulus generalization theory and the halo effect in sociology [177] and cognitive science [168], we propose a novel approach to predict user preferences by learn-

ing from both observed text and missing text based on the missing not at random hypothesis. We build a learning model which capable of using both observed data and missing text to conduct domain specific preference prediction. Because we discover the problem and evaluate our proposed model in on-line discussion forums, where the user text actually means user comments, hence we use the term comment to refer text in the rest of this subsection. First, we extract user-to-word and word-to-word relationships based on observed comments to obtain user and word representation. Then, we learn hidden user-word relationship based on the representation similarity between missed words and observed words. Similarly, the hidden user-to-user relationship is also built based on user representation. This two steps aim to model user's inclination even though they have missing comments and inclination. Thirdly, we consolidate the information from both hidden user-to-word matrix and the observed user-to-word matrix. We obtain the final user representation through joint matrix factorization of both the User-User similarity matrix and User-Word matrix. The factorized user vector is fed into a soft-max layer for user preference prediction. Experiments on our collected Chinese Hu-pu user preference corpus show that by incorporating missing user comments, our model is able to outperform the current models.

5.2.1 Proposed model

The diagram of our proposed model is shown in Figure 5.1. First, the raw data are used as the observed comments in the user and word integrated heterogeneous network to learn embedding representations of users and words, as U and W respectively. Based on these representations, we then construct our Hidden User-Word Matrix(H) as well as the User-User similarity matrix(I). We also construct the Observed User-Word Matrix(X) from the raw data. We build the Consolidated User-Word Matrix(F) by linear composition of both the Observed User-Word Matrix and the Hidden User-Word Matrix. Finally, we apply joint weighted matrix factorization to obtain the Final User Representation to be fed into

a soft-max classifier for preference prediction. We refer to this proposed model as the Combined Hidden Comments with Joint matrix Factorization (CHCJF).

Embedding representations of Users and Words

There are two kinds of relations from the observed comments which we consider useful for user preference identification: (1) User-to-word makes up a selection network and (2) Word-to-word forms a co-occurrence network. These two relations reflect the homophily relations and thus are useful in preference prediction [67, 219]. These two kind of relations can be modeled in one heterogeneous network as network embedding. We refer to this subjectivity-based modeling method as User-Word Heterogeneous Network Embedding (UWHNE). This module will describe the similarity between 'silent' comments and observed comments for modeling user inclination.

Let G denote a heterogeneous network where $G =< V, E >$; V is the set of notes representing either a user or a word; E is the set of edges between vertices; each edge $e \in E$ represents an ordered pair $e = (u, v)$ associated with a weight $w_{uv} > 0$ to indicate the strength of the relation. In the rest of our paper, we use $W = \{w_1, w_2, ..., w_m\}$ to represent all words in observed comments, $U = \{u_1, u_2, ..., u_n\}$ to represent all users. Thus $U \cup W = V$.

- Word-to-word relation: If two words appear in a context within a window of size k, we consider them to have a word-to-word relation with a link in G. The weight of the edge is based on the co-occurrence of the two words within the context windows. k is an algorithm parameter to be set experimentally.

- User-to-word relation: if a word is used by a user in his posting/comments, we consider them to have a user-to-word relation with a link in G. The weight of the edge is based on the number of times the word is used. The corresponding matrix for this relation is labeled as the Observed User-Word Matrix X in Figure 5.1.

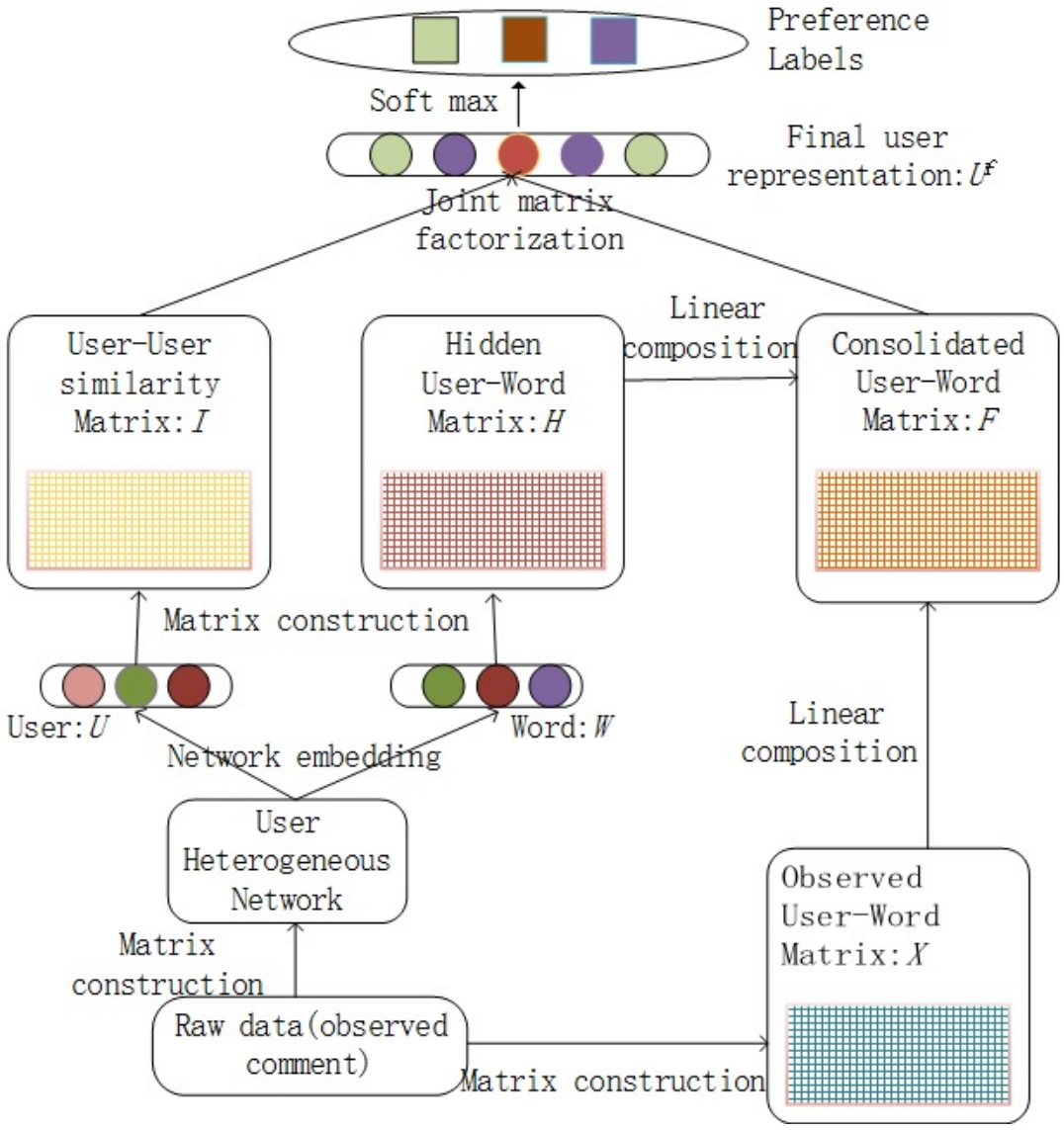


Figure 5.1: The overall system architecture of UWHNE model

The user representation and word representation are obtained using network embedding model inspired by Tang et al.[219]. For any vertex v_i and its representation U_i in the vector space, where U_i is a real number, let u'_i generally denote the neighbors of v_i, and let j. The conditional probability $p(v_j|v_i)$ can be defined by the soft-max function:

$$p(v_j|v_i) = \frac{exp(u'^T_j)}{\sum_V^{k=1} exp(u'^T_k * u_i)}. \tag{5.1}$$

To learn user representations and word representations using embedded vector learning, we make the conditional distribution of $p(v_j|v_i)$ to be close to its empirical distribution. The objective function O of a representation learning algorithm can be defined as:

$$O = -\sum_{(i,j)\in E} Weight_{ij} * logp(v_j|v_i). \tag{5.2}$$

Consequently, the vectors as representations for each user and each word is obtained. We can further construct the User-User Similarity Matrix, denoted by I, using cosine similarity.

Let U_i and U_k denote the representations of any two users u_i and u_k, the similarity between u_i and u_k can be defined as:

$$I_{i,k} = \frac{U_i^T U_k}{\|U_i\|_2 \|U_k\|_2}. \tag{5.3}$$

Construction of the Hidden User-Word Matrix

In this step we need to construct the Hidden User-Word matrix to model user inclination in missing comment. Let W_{u_i} denote all the observed words of user u_i, $W_{u_i} = \{w_{ui_1}, w_{ui_2}.....w_{ui_n}|w_{ui_j} \in W\}$, According to the stimulus generalization theory, if a word is close to the word which a user mentioned in his comments, it is more likely the word will be used by the user. We model this theory through the bellowing function:

- The Similarity of the observed words of user u_i and a hidden word w_j can either be calculated as the average or the maximum over all observed words. Below we only

give the definition based on average as:

$$S(W_{u_i}, w_j) = \frac{1}{n} \sum_{k=1}^{n} cos(w_j, w_{u_{i_k}}). \tag{5.4}$$

- For any user u_i and a hidden word w_j, we can then construct the Hidden User-Word Matrix H. $H_{i,j}$ represents the relationship between a user u_i and a hidden word w_j. Each element Hi, j in H is defined by

$$H_{i,j} = \left\{ \begin{array}{ccc} 0 & if & X_{i,j} \neq 0 \\ S(W_{u_i}, w_j) & if & X_{i,j} = 0 \text{ and } S(W_{u_i}, w_j) \geq \text{T} \\ \alpha & if & X_{i,j} = 0 \text{ and } S(W_{u_i}, w_j) \lneqq \text{T} \end{array} \right\}, \tag{5.5}$$

where T is the similarity threshold for filtering out words, $X_{i,j}$ is the value of user u_i to word w_j in the Observed User-Word Matrix. The strength of the relationship is determined by the similarity between the hidden word and its observed words. Some cells in H are filtered out by T which is determined by the distribution of similarity values. For hidden words, if the similarity to the observed words of users is below the threshold T, we give them a smoothing weight α.

Final User Representation by Joint Matrix factorization

In order to learn from user inclination in both the hidden and observed words, we first merge them to form the Consolidated User-Word Matrix, denoted by F. Let X denote the Observed User-Word Matrix. We merge the two matrices by linear composition and thus each $F_{i,j}$ is defined as:

$$F_{i,j} = \beta H_{i,j} + \gamma X_{i,j}, \tag{5.6}$$

where β and γ are the coefficients in the linear composition.

According to the homophily theory, a user is also related to other users who are similar to them. Thus, we model the Final User Representation using the joint weighted matrix

factorization by including both the User-User Matrix I and the Consolidated User-Word Matrix F.

Our goal is to find a user vector U^f such that for each u_i there is a $x_{u_i} \in R^f$, a vector $y_{w_j} \in R^f$ for each w_j, and a vector $z_{u_k^*} \in R^f$ for each factor in the User-User Similarity Matrix. These vectors are referred to as the user factor, the word factor and the user feature factor. If factor in $F_{i,j}$, the model should account for all possible user and words.

The factors are computed by minimizing the following objective function:

$$min_{x*y*}(\sum_{u_i,w_j}(F_{i,j} - x_{u_i}^T y_{w_j}) + \lambda(\sum_{u_i}\|x_{u_i}\|^2 + \sum_{w_j}\left\|y_{w_j}\right\|^2)). \tag{5.7}$$

The item $\lambda(\sum_u \|x_{u_i}\|^2 + \sum_i \left\|y_{w_j}\right\|^2)$ is a regularization item which aims to avoid over-fitting.

When we are building the User-User Similarity Matrix, the idea is similar. The purpose is to derive a high quality l dimensional feature representation $x_{u_i} \in R^f$ for each user u_i, and factor-specific latent feature vectors $z_{u_k} \in R^f$. Thus, the objective function is defined similarly:

$$min_{x*z*}(\sum_{u_i,u_k}(I_{i,k} - x_{u_i}^T z_{u_k^*}) + \lambda(\sum_{u_i}\|x_{u_i}\|^2 + \sum_{u_k^*}\left\|z_{u_k^*}\right\|^2)). \tag{5.8}$$

To reflect a user's relationship with other users and the user's preference to words together in the user's preference, we fuse both objective functions into a joint framework given below:

$$\begin{aligned} L(x,y,z) = min_{x*y*z*}(&\sum_{u_i,w_j}(F_{i,j} - x_{u_i}^T y_{w_j}) + \sum_{u_i,u_k^*}(I_{i,k} - x_{u_i}^T \\ &z_{u_k^*}) + \lambda(\sum_{u_i}\|x_{u_i}\|^2 + \sum_{w_j}\left\|y_{w_j}\right\|^2 + \sum_{u_k^*}\left\|z_{u_k^*}\right\|^2) + C), \end{aligned} \tag{5.9}$$

where C is a constant that does not depend on the parameters. Minimizing the loss function over two latent features with hyper-parameters equals to finding local minimum of the objective function by performing stochastic gradient descent in x_{u_i},y_{w_j} and $z_{u_k^*}$. The Optimized x_{u_i} is taken as the final User Representation, ${U_i}^f = x_{u_i}$.

User preference prediction

Let U^f denote the final user representation to be fed into a classification model. As the choice of classifiers is not the focus of this work, we simply take a commonly used soft-max classifier implemented by Liblinear [29]. Let K denotes the number of different labels for user preferences.

For a user u_i with the corresponding representation ${U_i}^f$, the predicted probability for for its label being j is defined by a soft-max function:

$$P(y = j|{U_i}^f) = \frac{e^{{U_i^f}^T \dot{W}eight_j}}{\sum_{k=1}^{K} e^{{U_i^f}^T \dot{W}eight_k}}, \tag{5.10}$$

where $Weight_j$ is the soft-max weight for each label.

Time complexity

The main computation of the gradient method is to evaluate the objective function L and its gradients against variables. Because of the sparsity of the matrix F, the computational complexity of evaluating the objective function L is $O(p_F + p_I)$, where p_F and p_I are the number of nonzero entries in matrices F and I. The computational complexity for each gradients x, y, z in the objective function is $O(p_F + p_I)$, $O(p_F)$ and $O(p_I)$, respectively. Hence, the computational complexity in each iteration is $O(p_F + p_I)$. This has linear complexity with respect to the number of non-zero value in the matrices. And the heterogeneous User-Word network could be optimized by edge sampling [219], according to Tang [219], the overall time complexity of network embedding is $O(dK|E|)$, where d a

constant and K is the number of negative sampling which is linear to the number of edges $|E|$.

5.2.2 Experiments

In the experiment part, we first introduce our works to build data from Hu-Pu dataset, then we compare our proposed CHCJF model with other representation model baseline, followed by effects of different embedding and factorization under CHCJF. Lastly, we introduce the effect of different parameters in the proposed CHCJF model.

Hu-Pu dataset

To evaluate our work, we need to source a data-set with preference labels. However, there is no ready public data to use. The data-sets used by most existing reported tasks are from some popular social websites like Facebook [3], which contain user generated text and preference in a structured format. However, these websites are usually publicly inaccessible, and a large portion of users tend to hide their profiles. In this work, we choose the publicly accessible basketball discussion forum Hu-Pu Basketball to obtain the dataset with naturally annotated gold labels for experiments.

HuPu Basketball is the biggest Chinese basketball discussion forum. Users can fill up profile information such as age, gender and location, etc., and they can choose one of the 20 CBA teams as their favorite-team. For data-set collection, we crawled all the discussion threads from March 2012 to April 2016. A total of 17,011 users clearly indicate their favorite-teams in their profile page with 423,758 observed comments. The statistics of user and words in their observed comments are listed in Table 5.3.

[3] https://www.facebook.com/

Statistics	User	Word
Overall Number	17,011	201,963
Min number of context	5	3
Max number of context	16,022	14,898
Ave number of context	344.40	58.51

Table 5.1: User statistics of experiment dataset

Table 5.2: Performance of different user representation models

	P	R	F
Doc2vec-F-only	0.4122	0.3774	0.3941
Doc2vec	0.4121	0.3838	0.3974
UWHNE-F-only	0.4281	0.3976	0.4123
UWHNE	**0.4299**	**0.4032**	**0.4161**

Comparison to Other Representations

We compare our algorithm with some of the state-of-art user representations in preference prediction including:

- Average word vector trained by domain specific data in HuPu(AW2V-HuPu);
- Average word vector trained by general domain Baidu Baike(AW2V-BaiKe);
- Latent Dirichlet Allocation (LDA), a widely used topic modeling model proposed by Blei [19];
- Singular Value Decomposition (SVD),a benchmark matrix factorization method proposed by Dumais et al.[53];
- Bag-of-Words(BOW), a text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity;
- TF-IDF(TFIDF), a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus;

- Document to vector representation (Doc2vec), an unsupervised algorithm that learns fixed-length feature representations from variable-length pieces of texts[113];

- Probabilistic Matrix Factorization (PMF), a low-dimensional factor model to perform matrix factorization by using probabilistic density function [64];

- Weighted matrix factorization (WMF): a model treating the data as indication of positive and negative preference associated with vastly varying confidence levels [68, 64].

- Neural Sentiment Classification with local attention, this model are regraded as the state-of-art model in document level sentiment analysis [33].

The same soft-max classifier is used for all user representation methods except BOW and TFIDF which use the SVM classifier. We run the experiment in 10-fold cross validation. The parameter setting for each model follows the same parameter used in the referenced works.

We label our Combined Hidden Comments with Joint Matrix Factorization model as CHCJF. The filtering threshold T set to 0.8 experimentally. The Hidden User-Word matrix H and the Observed User-Word matrix have the same weight in the composition($\beta = \gamma = 1$). The size of dimension are set to be 300 while the number of iterations is set to 20.

Note that in Table 5.3 LDA, AW2V-BaiKe, and Aw2v-HuPu show the worst results. This indicates that topics based representation and average word embedding are not good for preference prediction. Comparing to BOW or TFIDF with the SVM classifier as the baseline, SVD achieves slightly higher precision, yet its recall is worse than BOW and TFIDF. The document embedding method, although achieves highest precision at 40.14%, the F-score is still no better than TFIDF using SVM. The PMF and WMF are two popular matrix factorization methods aimed at addressing the missing data problem. These two models outperform all other reference models except our proposed model. Our proposed

	P	R	F
AW2V-Baike	0.1882	0.2481	0.2141
AW2V-HuPu	0.2181	0.2801	0.2453
LDA	0.1962	0.2623	0.2245
SVD	0.3657	0.3148	0.3383
BOW	0.3237	0.3437	0.3334
TFIDF	0.3638	0.3829	0.3726
Doc2vec	0.4014	0.3487	0.3707
PMF	0.3876	0.3864	0.3871
WMF	0.3942	0.4003	0.3972
NSC+LA	0.4178	0.3933	0.4051
CHCJF-UWHNE	**0.4299**	**0.4032**	**0.4161**

Table 5.3: Comparison experiments

	Doc2vec	UWHNE
P-value in P	0.3652	0.1382
P-value in R	0.0280	0.0257
P-value in F	0.0632	0.0672

Table 5.4: Significance by adding User-User Similarity Matrix

model outperform over the state-of-art WMF model by 1.89% in F-score with the the P-value at $8.5\text{e}{-13}$. The model even have better performance than the state-of-art document level classification model on this sparsely user activities data set. This proves that giving different weights based on the similarity between missing words and observed words helps to boost the performance quite significantly. The set of experiments not only proves that making use of predicted missing comment can help to improve user preference predictions, the way the missing comments are incorporated into the prediction also matters a lot.

Effects of Different Embedding and Factorization under CHCJF

Table 5.4 shows the performance of the four variants under our CHCJF model. As we shown in the table, the Doc2vec model achieves comparable results compare to the WMF model in F-score. But, it under-performs significantly compared to the UWHNE models, the F-score is 1.87% lower than the UWHNE model with the P-value of $9.2\text{e}{-8}$. This

is because user-word heterogeneous network embedding model is able to scale to very large, arbitrary types of networks: the directed user-to-word network and the un-directed word-to-word network. This model optimizes an objective which can preserve both the local and the global network structures [219]. The doc2vec model does not provide a clear objective to preserve network properties as it only focuses on the co-occurrence relationship between users and words. Comparing to the use of joint matrix factorization, we note that the Doc2vec-F-only model and the UWHNE-F-only model under-perform by 0.33% and 0.39% in F-score, respectively. This indicates that including User-User similarity information(I) can further improve prediction performance.

Table 5.4 shows the effect of the User-User Similarity Matrix on both Doc2vec and UWHNE in P-values. Although the improvement in precision is considered insignificant, the P-value in recall is well below 0.05, which means the improvement in recall brought by User-User Similarity Matrix is significant. The P-value in F-score are both well below 0.1 but slightly above 0.05. This indicates that the use of joint matrix factorization, incorporating homophily theory in the User-User Similarity Matrix, can help to boost the performance in a significant level.

Effects of hyper-parameters

Two hyper parameters can affect the performance of our algorithm. The first one is the similarity threshold T for the Hidden User-Word Matrix H. The second one is the iteration number of joint matrix factorization.

Figure 5.3 shows the effect of iteration number to F-score at 10, 20, 40, 50, 75, and 100 while keep the other parameters fixed. Note that the performance is quite stable when the iteration number is in the range of 40-80. Although, we do see some changes with the iteration number, the variation is stablized at around 0.4163 with $\pm$ 0.004 change in F-score. This is because the change of precision is negatively correlated to recall. This means that our method is not sensitive to the number of iterations. Table 5.2 shows the

effect of threshold T to P, R and F in 10-fold validation. We can easily observe that the choice of T has significant impact on the model.The performance will increase with T first when $T = 0.8$ gives the best performance, then the performance decrease with T continue to raise. A low threshold can bring noise into the model while a high T will not be able to include sufficient information on missing words.

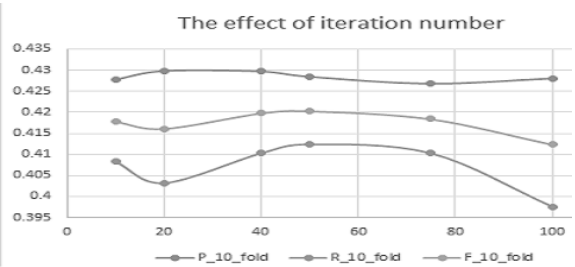


Figure 5.2: The effect of iterations in CHCJMF-UWHNE

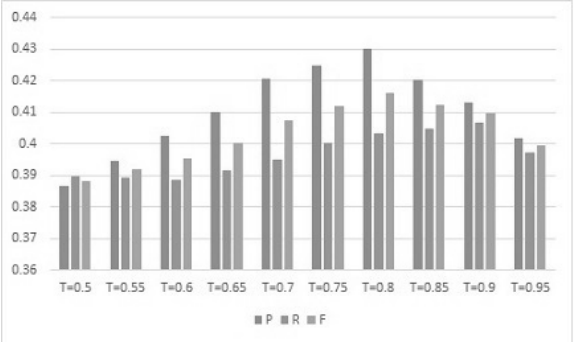


Figure 5.3: The effect of T in CHCJMF-UWHNE model

5.2.3 Conclusion on learning user profile from observed text and missing text

Many user profile prediction models follow the missing at random hypothesis. Thus only observed comments are used. However, the stimulus generalization theory and the halo effect suggest that missing comments can be used together to determine a user's preference. Inspired by the findings in sociology and cognitive science. we propose a novel approach to learn user preferences from both observed comments and missing comments in a heterogeneous network embedding with joint matrix factorization.

Performance evaluation using a soft-max classifier for user preference prediction shows that our proposed model makes 1.89 % improvement in F-score compared to the current start-of-art representation methods with p-value of 8.5e-13 to indicate the improvement is very significant. Our experiments prove that missing comments does not follow the missing at random hypothesis and user inclination can still be learn even if they are mostly

silent.

Experiments also show that an appropriate way to obtain user representations from observed comments help to improve performance. Compare to Doc2vec, our proposed UWHNE can obtain more reliable user representation and makes 1.87% improvement in F-score with the P-value of $9.2\mathrm{e}{-8}$. Our learning method does not need to reply on any external knowledge. This is particularly important for resource poor languages and for informal text written in social media. Future research can explore more fine grained relations between missing words and observed words.

5.3 Learning user profile from text and Links

This section introduce our works in learning user embedding from both text and links information. Generally speaking, there are two types of networks: homogeneous networks and heterogeneous networks. A homogeneous network only consists of one type of nodes and behavior. A heterogeneous network contains different types of nodes. In many heterogeneous networks, especially user related networks, users (the subjects of behaviors) and products (the objects of behaviors) are commonly regarded as two different types of nodes. Both text and links can exist in homogeneous network and heterogeneous network, our model are designed for both homogeneous network and heterogeneous network.

To integrate link structure and text in the same network, two main issues need to be addressed: the first issue is how to learn node representation by integrating link information and text content coherently; the second issue is how to distinguish different types of nodes in the representation framework. This is particularly difficult if nodes are comprehensively related yet are of different types.

In this work, we propose a novel method to learn node representation in heterogeneous networks through the learnings of both link structure and available text content in a unified framework. To learn node representation, we sample each type of links sepa-

rately to obtain conditional probabilities, and then the sampled edges are treated as binary links for model updating. This embedding-based link learning method is derived from the random walk method proposed by Tang et al.[219]. For text embedding, we propose to measure conditional probabilities of both link information and text information between any two nodes. We also propose a two-step neural network to process text not only at sentence level (individual user and product related comments), but also at document level (the collection of user and product related comments). This way, we can obtain more comprehensive information including attentions and global semantics. A Recurrent Neural Network (RNN) model is used to assemble sentence level information. Furthermore, an attention based Convolutional Neural Network (CNN) model is used to extract sentence-to-document level information more effectively. For the evaluation of heterogeneous networks, a large-scale heterogeneous network dataset is collected for embedding learning and will be made available for public access.

The contribution of this chapter includes:

- A novel neural network based node representation model in a joint learning framework. This framework incorporates both structured link information and unstructured text information in a hierarchical neural network. It has the capability to learn multiple types of nodes in a heterogeneous network.

- A novel hierarchical neural network model to obtain network embedding of text to include both sentence level information and document level information.

- A novel attention mechanism by extending the text of adjacent nodes through linked edges so that much larger context in the network can be included.

- Provision of an open accessed heterogeneous network dataset.

Evaluations on the link prediction in four benchmark datasets shows significant performance boost compared to state-of-the-art methods.

The rest of the chapter is organized as follows: Section 5.3.1 introduces our proposed method for joint learning of heterogeneous network embedding from links and text. Section 5.3.2 elaborates on the evaluation of several network embedding datasets to validate the effectiveness of our proposed method. Section 5.3.3 concludes this paper with future direction.

LQ Notes: Why only in this part you have this.

5.3.1 Proposed model

Network nodes often have both link and text content regardless of its homogeneity. For easy explanation of the formalism used in this paper, we first introduce an animation video website Bilibili [4] as an example to demonstrate network heterogeneity and how text can be used in different perspectives. The Bilibili website has two types of nodes. User nodes, as one type of nodes, is generally involved with a friend/follower network. Text comments written by users are posted on a bulletin linking to specific animation videos. The collection of reviews by a particular user is a good source to find information about the user's personal preferences in addition to his/her subjective opinions of the videos. Video nodes, another node type related to animation, also have a collection of comments written by different users. The collection of reviews by different users for a given video reflect the collective opinions and should be more objective on the whole.

In general, a heterogeneous network G can be represented as a graph $G = (V, E, T)$, where V is the set of nodes, E is the set of edges, and T is the set of documents. Furthermore, V can be of different types. For easy illustration without loss of generality, let us assume a heterogeneous network has two types of nodes: user nodes and product nodes [5], denoted by u_i and a_k such that $u_i \in V_u$ and $a_k \in V_a$ and $V_u \cup V_a = V$. The two types of nodes are connected by two types of links (edges): user-to-user links ($e_{uu} =< u_i, u_j >$)

[4] https://www.bilibili.com/

[5] In Bilibili, products are essentially animation videos, hence we use a_k to represent product nodes. In other datasets, nodes can have different names but the process model should be the same.

and user-to-product links ($e_{ua} =< u_i, a_k >$). Since there is no direct connection for videos in Bilibili, product-to-product links are omitted.

The general aim of network node embedding is to learn a low-dimensional vector representation $\vec{v} \in R^d$ for each node according to links and associated node information. Note that the dimension size d=$|\vec{v}|$ of vector $\vec{v}$ is much smaller than $|V|$, the size of the network.

We propose a novel node representation learning method which jointly learn Link and Text Embedding for Heterogeneous network nodes (LTEH). Here, the term link embedding means to learn node embedding based on link information only. Let $\vec{v}^l$ denote link embedding and $\vec{v}^t$ denote text embedding, respectively. Then, the node representation $\vec{v}$ can be obtained by a weighted concatenation of link embedding and text embedding $\vec{v} = \alpha * \vec{v}^l \oplus \beta * \vec{v}^t$, where α and β can be learned though an optimization process.

The objective of LTEH is to obtain optimized node representation by making use of both link and text information. The overall loss function $L(e)$ of all links $e \in E$ is formed by the summation of the network link loss function $L_l(e)$ and the text loss function $L_t(e)$ in a jointly optimized approach defined below:

$$L = \sum_{e \in E} (L_l(e) + L_t(e)). \tag{5.11}$$

Link embedding

Since our heterogeneous network contains two types of nodes, the loss function for link embedding $L_l(e)$ in Formula 5.12 should consider both types of nodes. Let u and a denote user and video nodes, respectively. Since the links of an individual user node can be two types, the e_{uu} type and the e_{ua} type, the loss function of a user should includes two items. On the other hand, a video is only associated with e_{au} links, thus its loss function should have only one item. Consequently, the loss function of link embedding $L_l(e)$, given below in Formula 5.12, is defined as the addition of two parts: The first part in square brackets is

the loss function of user embedding with two link-type probabilities and the second part is the loss function of video embedding with one link-type probability:

$$L_l(e) = \left[W^l_{uu} log(p_l(u_i|u_j)) + W^l_{ua} log(p_l(a_k|u_i))\right] + W^l_{au} log(p_l(u_i|a_k)), \quad (5.12)$$

where W^l_{uu}, W^l_{ua}, and W^l_{au} are the weight parameter vectors for the three types of links: user-to-user(uu), users-to-(animation) videos (ua), and (animation) video-to-user (au), respectively, the superscript l stand for link embedding. Formula 5.12 shows that two different types of nodes are represented differently. A user node can connect to both other users and animation videos so its conditional probability has two components defined by the addition of Formula 5.13 and Formula 5.14 as the first two elements in Formula 5.12. Since a video node only has one type of links, its conditional probability is defined by Formula 5.15 only. The conditional probabilities of the three types of links are listed below:

$$p_l(u_i|u_j) = \frac{exp(\vec{u}_i.\vec{u}_j)}{\sum exp(\vec{u}_i.\vec{V})}, \quad (5.13)$$

$$p_l(a_k|u_i) = \frac{exp(\vec{a}_k.\vec{u}_i)}{\sum exp(\vec{u}_i.\vec{V})}, \quad (5.14)$$

and

$$p_l(u_i|a_k) = \frac{exp(\vec{u}_i.\vec{a}_k)}{\sum exp(\vec{a}_k.\vec{V})}, \quad (5.15)$$

where V refers to all nodes in the network.

Text based objective function

The loss function of text embedding should consider text in association with three types of links low in Formula 5.16, similar to discussion on link embedding:

$$L_t(e) = L_t(u_i, u_j) + L_t(u_i, a_k) + L_t(a_k, u_i). \quad (5.16)$$

For an e_{uu} link, its loss function is defined by three components:

$$L_t(u_i, u_j) = \alpha_1 L_{tt}(u_i, u_j) + \beta_1 L_{tl}(u_i, u_j) + \gamma_1 L_{lt}(u_i, u_j). \quad (5.17)$$

In Formula 5.17, $L_{tt}(u_i, u_j)$ is the loss between text embedding of an e_{uu} link. $L_{tl}(u_i, u_j)$ is the loss between the text embedding of user u_i and the link embedding of user u_j. $L_{lt}(u_i, u_j)$ is the loss between the link embedding of user u_i and text embedding of user node u_j. $\alpha_1, \beta_1, \gamma_1$ are three weighted parameters for the three loss functions. We optimize the conditional probabilities for all the vector representations in Formula 5.17 as:

$$L_{tt}(u_i, u_j) = W_{uu}^{tt} log(p_{tt}(u_j|u_i)), \quad (5.18)$$

$$L_{tl}(u_i, u_j) = W_{uu}^{tl} log(p_{tl}(u_j|u_i)), \quad (5.19)$$

and

$$L_{lt}(u_i, u_j) = W_{uu}^{lt} log(p_{lt}(u_j|u_i)), \quad (5.20)$$

where W_{uu}^{tt}, W_{uu}^{tl}, and W_{uu}^{lt} are weighted matrices. Similarly, the last two elements in the loss function given in Formula 5.16 can be defined as:

$$L_t(u_i, a_k) = \alpha_2 L_{tt}(u_i, a_k) + \beta_2 L_{tl}(u_i, a_k) + \gamma_2 L_{lt}(u_i, a_k), \quad (5.21)$$

and

$$L_t(a_k, u_i) = \alpha_3 L_{tt}(a_k, u_i) + \beta_3 L_{tl}(a_k, u_i) + \gamma_3 L_{lt}(a_k, u_i). \quad (5.22)$$

$\alpha_2, \beta_2, \gamma_2$ are three heterogeneous weights for the three loss functions in function 5.21, and $\alpha_3, \beta_3, \gamma_3$ are three weighted parameters for the three loss functions in Formula 5.22. Probability functions map both link embedding and text embedding onto the same representation space. Softmax function is used to obtain all the probabilities. Now, the main task is to obtain text embeddings of nodes.

Text embedding

Most text embedding models examine the context of words at sentence level, which is considered a shallow approach. A more comprehensive approach is to consider the collection of sentences for a node to include other information such as attentions and statistics at a macro level. The main idea in this work treats a collection of sentences in a node as one document to perform embedding in both sentence level and document level. We propose to use a hierarchal neural network to obtain comprehensive semantic information by capturing aggregated word information at sentence level in the first layer, and then capturing aggregated sentence information at document level in the second layer.

Let T be the collection of documents associated with n nodes: $T = T_1...T_i...T_n$. The text T_i for node i is made up by a series of sentences: $T_i = S_1...S_j...S_{l_i}$ where l_i is the number of sentences in T_i. A sentence S_j is made up of a sequence of words $S_j = {w_1}^j...{w_k}^j, {w_{l_j}}^j$ where l_j is set to be the length of S_j. Each word ${w_k}^j$ is initialized as a fixed dimension vector $\vec{w}_k^j \in R^d$, where d is the size of word vectors.

At the sentence level, sentence embedding using neural networks is learned by three layers:

- Look up layer: Given a word ${w_k}^j \in S_j$, and $S_j \in T_i$, this layer transforms each word into its word embedding.
- Recurrent layer (RNN): Each cell in this layer runs from the first word in the sentence to the last word. For a sentence S_j with length l_j, the RNN architecture is an

l_j sequential network. The original word vector series $\vec{w}_1^j, \vec{w}_2^j ... \vec{w}_n^j$ are transfered to d dimensional hidden vectors through recurrent cells: $\vec{h}_1^j, \vec{h}_2^j ... \vec{h}_l^j$. The output is a matrix of size $l_j * d$.

- Average pooling: This layer is used to obtain the embedding of S_j ($S_j \in T_i$). Average pooling with non-linear transformation is defined as follows:

$$\vec{S_j} = tanh(avg(\vec{h}_1^j, \vec{h}_2^j ... \vec{h}_1^j)). \tag{5.23}$$

At the document level, LTEH uses an attention based CNN model. This can give higher weights to the more salient sentences in the collection. The CNN model with attention mechanism consists of three layers:

- Convolution layer: This layer extracts sentence-to-document level information. For a $T_i \in T$ with n sentences $T_i = S_1, S_2 ... S_n$, we perform convolution operation over a window of size l by using a convolution matrix $C \in R^{l*(l*d)}$ defined by:

$$\vec{x}_i = C_i S_{i:i+l-1} + b, \tag{5.24}$$

where $S_{i:i+l-1}$ denotes the concatenation of sentence embeddings learned from the sentence level with window size of l. b is a regularization parameter.

- Attention layer: Attention weights are learned from text content of both types of nodes in a link. For an e_{uu} link $< u_i, u_j >$, let the corresponding vector outputs from the convolution layer be $\vec{x}_1^i ... \vec{x}_h^i ... \vec{x}_n^i$ and $\vec{x}_1^j ... \vec{x}_o^j ... \vec{x}_m^j$, respectively . The attention weights for each word ${x_o}^i$ and ${x_o}^j$ are defined by:

$$W_{x_h^i} = \frac{\sum_{o=1}^{m} \vec{x}_h^i \vec{x}_o^{j^T}}{Z}, \tag{5.25}$$

and

$$W_{x_o^j} = \frac{\sum_{h=1}^{n} \vec{x}_o^j \vec{x}_h^{i^T}}{Z}, \tag{5.26}$$

where

$$Z = \sum_{h=1}^{n} \sum_{o=1}^{m} \vec{x}_h^i \vec{x}_o^{j^T} + \sum_{o=1}^{m} \sum_{h=1}^{n} \vec{x}_o^j \vec{x}_h^{i^T}. \tag{5.27}$$

The attention weight of word x_h, represented by $W_{x_h^i}$, is calculated from the vector production between its own representation $\vec{x}_h$ and every word in the content of linked user u_j, noted as $\vec{x}_o$. The attention weight of word x_o, represented by $W_{x_o^j}$, is calculated from the production between its own representation $\vec{x}_o$ and every word in the content of linked user u_i, noted as $\vec{x}_h$.

- Pooling: This layer assembles sentence vectors and attention weights into document representations for $\vec{u}_i^t$ and $\vec{u}_j^t$ (the superscript mark t stands for text embedding) as:

$$\vec{u}_i^t = \sum_{h=1}^{n} W_{x_h{}^i} * \vec{x}_h^i, \tag{5.28}$$

and

$$\vec{u}_i^t = \sum_{o=1}^{m} W_{x_o{}^j} * \vec{x}_o^j. \tag{5.29}$$

Functions 5.25, 5.26, and 5.27 indicate that in our proposed LTEH model, the attention weight for a sentence S_j is not only determined by its document context, but also text extracted through the linked nodes. Similarly, for e_{ua} links, the attention mechanism can also be obtained. Because user and video have different text content, our model has the ability to capture the differences between the two types of links.

Time complexity discussion

The time complexity of link embedding in our work is basically the same as the LINE model given in Tang's work on large-scale information link embedding (LINE)[219]. For text embedding, a two-stage processing architecture is used. For a network with $|V|$ nodes and $|E|$ edges, let us assume that each node has $|l|$ sentence and each sentence has $|m|$ words. Then, the sentence level RNN has a complexity of $O(|m|)$. The document level CNN has a complexity of $O(|l|)$, and attention mechanism has a complexity of $O(|E|)$. Thus, the overall time complexity of text embedding is $O(|m||l|(|V| + |E|))$.

5.3.2 Experiments

To evaluate the effectiveness of our proposed LTEH model, we conduct two downstream tasks. The first task is on link prediction. The second task is on node classification. We also use data visualization method to compare our model with CANE-A, the state-of-the-art model.

Evaluated systems

Three groups of algorithms are used for performance evaluation and comparison[6]. Group One has three baseline algorithms that only use network link information including:

- **Deepwalk** [180], a model using local information obtained from truncated random walks to learn latent representations by treating walks as the equivalent of sentences.
- **LINE** [219], a traditional network model using both first-order and second-order proximity in network graph.
- **Node2vec** [66], a random walk based method to sample neighbor nodes.

Group Two algorithms use both link information and text information without the use of attention mechanism including:

[6] In the table, three groups are named as G1, G2, G3.

- **TADW** [250]: a state-of-the-art algorithm using both network and text information to learn node representation.
- **TriDNR** [172]: a tri-party deep network algorithm that exploits links, node content, as well as label information.
- **CENE** [225]: a method to simultaneously detects community distribution of each node and learns the embeddings of both nodes and communities jointly.
- **CANE-N** [224]: a contrast method by the state-of-the-art CANE system to obtain node embedding through context-aware embedding without attention mechanism.
- **LTEH-N**: a variation of our proposed LTEH, LTEH-N refers to the LTEH model without including the attention mechanism.

Group Three includes two models with built in attention models:

- **CANE-A**: the state-of-the-art attention based network embedding model designed for learning embedding from link and content, proposed by Tu et al [224].
- **LTEH-A**: our full LTEH model with document level processing and attention mechanism.[224].

For easy comparison, the network embedding sizes of LTEH-N and LTEH-A are set to 200 as all relevant models in the evaluation dimension size of 200.

Datasets

Table 5.5 lists the four datasets used for performance evaluation. The datasets are divided into two groups. The first group has three homogeneous networks datasets used in Tu et al's work [224] with text content including Cora, Hepth, and Zhihu. Details of the three datasets are listed below:

- Cora [7]: A typical citation network dataset [147] consists of 2,708 scientific papers as network nodes (belong to 7 categories) 5,429 links on authors. Cora is the only dataset which has isolated nodes (66 in total) and thus they have no network behavior. This is the first homogeneous network datasets proposed in network embedding.

- Hepth [8] (High Energy Physics Theory): A citation network originally from arXiv3 [117]. This collection is the exact version used by Tu's work [224]. The network has 1,038 paper nodes and 1,990 identical author links.

- Zhihu[9]: An on-line Q&A website in China where users can follow each other and answer questions on this site. The dataset has 10,000 active users and 43,894 links from Zhihu [224]. Punctuations of the text are removed. Thus, there is no document structure.

Data	Nodes	Edges	Words /node	Sent. /node	Total text
Cora	2,277	5,214	90.44	5.32	206K
Hepth	1,038	1,990	54.47	3.26	56K
Zhihu	10,000	43,894	89.61	N/A	896K
Bili_user	3,401	4,259	305.61	12.63	1.039M
Bili_video	1,433	N/A	725.31	29.97	1.039M
Bili_all	4,834	13,801	430.02	17.78	2.079M

Table 5.5: Statistics of four benchmark datasets

The second group has one heterogeneous dataset collected by this work for benchmarking. The data is from the Bilibili website, referred to as Bilibili. Bilibili is an animation video sharing website of anime, manga and game fandom based in China. Users can submit, view, and add comments on products. In the Bilibili dataset, products are essentially animation videos. Hence, nodes in Bilibili are either user nodes or animation video nodes. The total number of e_{uu} links is 4,259. The total number of e_{ua} links is 9,542. Thus, the

[7] https://relational.fit.cvut.cz/dataset/CORA

[8] https://snap.stanford.edu/data/cit-HepTh.html

[9] https://www.zhihu.com/

total number of links is 13,801. The Bilibili dataset contains 3,400 users and 1,434 animation videos to form a heterogeneous network with a total of 4,834 nodes. The statistics of the four datasets are listed in Table 5.5.

Evaluation of link prediction

	Model	Dataset	15%	25%	35%	45%	55%	65%	75%	85%	95%
G 1	Deepwalk†	Cora	56.0	63.0	70.2	75.5	80.1	85.2	85.3	87.8	90.3
		Hepth	55.2	66.0	70.0	75.7	81.3	83.3	87.6	88.9	89.0
	LiNE†	Cora	55.0	58.6	66.4	73.0	77.6	82.8	85.6	88.4	89.3
		Hepth	53.7	60.4	66.5	73.9	78.5	83.8	87.5	87.7	87.6
	Node2vec†	Cora	55.9	62.4	66.1	75.0	78.7	81.6	85.9	87.3	88.2
		Hepth	57.1	63.6	69.9	76.2	84.3	87.3	88.4	89.2	89.2
G 2	TADW†	Cora	86.6	88.2	90.2	90.8	90.0	93.0	91.0	93.4	92.7
		Hepth	87.0	89.5	91.8	90.8	91.1	92.6	93.5	91.9	91.7
	TriDNR	Cora	85.1	87.9	88.3	89.1	90.7	90.4	92.2	93.3	94.1
		Hepth	87.5	88.4	89.8	90.4	91.5	91.7	91.9	92.5	92.3
	CENE†	Cora	72.1	86.5	84.6	88.1	89.4	89.2	93.9	95.0	95.9
		Hepth	86.2	84.6	89.8	91.2	92.3	91.8	93.2	92.9	93.2
	CANE-N†	Cora	85.8	90.5	91.6	93.2	93.9	94.6	95.4	95.1	95.5
		Hepth	84.5	89.3	89.2	91.6	91.1	91.8	92.3	92.5	93.6
	LTEH-N†	Cora	82.0	84.9	88.8	89.5	90.3	90.9	91.3	93.7	94.2
		Hepth	86.9	88.0	87.1	90.1	90.0	91.7	91.9	94.1	***95.8***
G 3	CANE-A†	Cora	**86.8**	**91.5**	**92.2**	**93.9**	**94.6**	**94.9**	**95.6**	**96.6**	**97.7**
		Hepth	90.0	91.2	92.0	93.0	94.2	94.6	95.4	95.7	96.3
	LTEH-A	Cora	83.5	86.5	90.4	90.8	92.3	92.9	93.3	94.5	95.4
		Hepth	87.9	88.5	88.9	90.7	90.8	92.9	93.4	96.1	96.8
	(p-value)	Cora	10^{-9}	10^{-11}	10^{-5}	10^{-5}	10^{-13}	10^{-6}	10^{-7}	10^{-5}	0.003
		Hepth	10^{-8}	10^{-8}	10^{-9}	10^{-9}	10^{-10}	10^{-7}	10^{-5}	0.040	0.010

Table 5.6: AUC results of the two small homogeneous datasets Cora and Hepth

For link prediction, we run the models using the three homogeneous datasets provided by Tu et al. [224] in addition to the Bilibili heterogeneous dataset. For fair comparison, we use the reported parameters provided by Tu et al. [224, 172] for previous works. Performance is measured by the commonly used AUC values (area under the ROC curve[10])[74].

[10] https://en.wikipedia.org/wiki/Receiver-operating-characteristic

	Model	15%	25%	35%	45%	55%	65%	75%	85%	95%
G 1	DeepWalk†	56.6	58.1	60.1	60.0	61.8	61.9	63.3	63.7	67.8
	LINE†	52.3	55.9	59.9	60.9	64.3	66.0	67.7	69.3	71.1
	Node2vec†	54.2	57.1	57.3	58.3	58.7	62.5	66.2	67.6	68.5
G 2	TADW†	52.3	54.2	55.6	57.3	60.8	62.4	65.2	63.8	69.0
	TriDNR	55.1	56.9	61.8	62.3	65.8	68.8	69.2	70.4	71.5
	CENE†	56.2	57.4	60.3	63.0	66.3	66.0	70.2	69.8	73.8
	CANE-N†	56.7	59.1	60.9	64.0	66.1	68.9	69.8	71.0	74.3
	LTEH-N	59.9	62.8	66.3	68.7	69.9	70.6	71.8	72.1	74.8
G 3	CANE-A†	56.8	59.3	62.9	64.5	68.9	70.4	71.4	73.6	75.4
	LTEH-A	**61.9**	**64.8**	**68.3**	**71.9**	**72.5**	**73.4**	**74.2**	**74.5**	**78.9**
	(p-value)	10^{-9}	10^{-9}	10^{-9}	10^{-11}	10^{-13}	10^{-15}	10^{-18}	10^{-21}	10^{-23}

Table 5.7: AUC results of the large homogeneous Zhihu dataset

	Model	15%	25%	35%	45%	55%	65%	75%	85%	95%
G 1	DeepWalk	51.6	54.1	55.1	55.6	56.8	56.9	57.3	57.7	59.8
	LINE	52.3	55.9	56.9	58.3	59.1	59.4	61.7	62.3	62.8
	Node2vec	54.2	57.1	57.3	58.3	58.7	62.5	66.2	67.6	68.5
G 2	TADW	55.3	56.2	56.6	57.1	60.8	62.4	63.5	64.9	65.8
	TriDNR	55.6	58.7	62.0	62.3	63.7	66.0	68.6	68.6	70.9
	CENE	56.6	59.9	60.4	63.3	64.8	67.9	68.9	69.6	71.0
	CANE-N	57.4	58.7	61.2	62.1	63.3	66.4	67.6	68.0	70.2
	LTEH-N	57.1	57.7	59.3	60.9	62.5	63.9	65.2	68.7	71.1
G 3	CANE-A	57.6	60.7	63.8	64.3	65.0	67.6	69.6	70.5	73.0
	LTEH-A	**60.1**	**62.1**	**64.2**	**66.2**	**68.5**	**69.9**	**71.5**	**75.6**	**78.0**
	(p-value)	10^{-12}	10^{-12}	10^{-12}	10^{-12}	10^{-12}	10^{-12}	10^{-9}	10^{-12}	10^{-25}

Table 5.8: AUC results of the heterogeneous datasets Bilibili

Experiments are conducted using different training/test ratios from 15% to 95% with 10% increase in each increment, and average AUC is used on five rounds of random tests. We also show the p-value (the result of t-test) by running our proposed LTEH-A model 10 times and compare it to the state-of-the-art model.[11] In the following tables, results marked by † are performance directly reported by related references.

[11] All p-values in the Table 5.6 and 5.7 is the results of t-test by comparing our proposed LTEH-A model with CANE-A model.

Table 5.6 shows the results on the two small homogeneous datasets, Cora and Hepth. Table 5.7 shows the performance of Zhihu. Bold font highlights the best result and the second best is highlighted by underline. Note that CANE-A which uses attention mechanism is indeed the best performer on both Cora and Hepth datasets. Both models of LTEH-N and LTEH-A do not show advantage over the state-of-the-art model in these two relatively small datasets even though LTEH-A does show much improved performance when higher percentage of data are used. The main reason is that LTEH requires more data for training regardless of homogeneity. On the Cora data, which has 3 % of isolated nodes without links, LTEH again cannot take advantage of its attention model as it needs to follow links to extend the context for attention model to work.

Our proposed LTEH-A starts to show its advantage when training data reaches 75%. This is because LTEH includes a sentence to document level embedding which requires more data for training. For models not using attention mechanism, no particular method has obvious advantage. However, Table 5.6 indicates that using text information has a clear advantage compared to methods using links only.

Table 5.7 shows the performance on the Zhihu dataset, which is larger than Cora and Hepth. Note that on the Zhihu dataset LTEH-A has a consistent performance improvement over other baselines including CANE-A. The range of improvement compared to CANE-A is from 0.9% to 7.4%. Table 5.7 also lists the *p-value*s of our model compared to the best state-of-the-art baseline (CANE-A) to indicate the significance of improvements. As the largest p-value is in the scale of 10^{-9}, this means that improvements in the whole range are very significant. Among all the methods without using attention mechanisms, our proposed LTEH-N also performs better than the state-of-the-art method CANE-N by as much as 5.4%.

Performance on the heterogeneous dataset Bilibili is shown in Table 5.8. Note that our proposed LTEH-A which uses an attention mechanism consistently outperforms other baselines in all training ratios including the state-of-the-art CANE-A system. The im-

provement ranges from 0.4% to 5.0%. This indicates that LTEH-A can make more effective use of context information for link prediction. In Bilibili, the largest *p-value* is in the scale of 10^{-9}. This means that improvements by LTEH-A are very significant in the whole data range.

To evaluate the effectiveness of document level embedding, let us now focus on algorithms in the second group. It is interesting to observe that even though Bilibili is rich in text information, none of the methods in this group has a clear advantage. Other than TriDNR, which consistently underperforms in this dataset, the best performers scatter among TADW, CENE, CANE-N, and LTEH-N. By a closer look, LTEH-N gives a good performance with improvement margin of at least 0.7% compared to all the other methods when the train data reaches 85% and 95%. In other words, even without the use of attention mechanisms, the aggregation of sentence information at document level still helps LTEH-N to outperform CANE-N and CENE when training data is sufficiently large. In Zhihu, LTEH-N outperforms all baseline methods including state-of-the-art CANE-N and CENE in all data range. The improvement range is from 0.5% to 5.4%.

Comparing LTEH-A with LTEH-N, we conclude that the attention mechanism using links and text plays a very important role in node representation. This can be seen from the fact that LTEN-A has achieved higher AUC value in all four datasets than LTEN-N. Obviously, richer text information with extended context helps to built a better attention model. In the Bilibili dataset, a node has far more number of words and sentences than that in the other three datasets. On average, LTEN-A is 2.0% to 6.9% higher in AUC in Bilibili (Table 5.8) than LTEN-N, compared to about 1.0 % improvement in Hepth and Cora (Table 5.6).

Evaluation of different heterogeneous weights

In each function of 5.17, 5.21, and 5.22, three parameters are used as weighted parameters. Firstly we evaluate the three weighted parameters in formula 5.17, $L_{tt}(u_i, u_j)$ is the loss

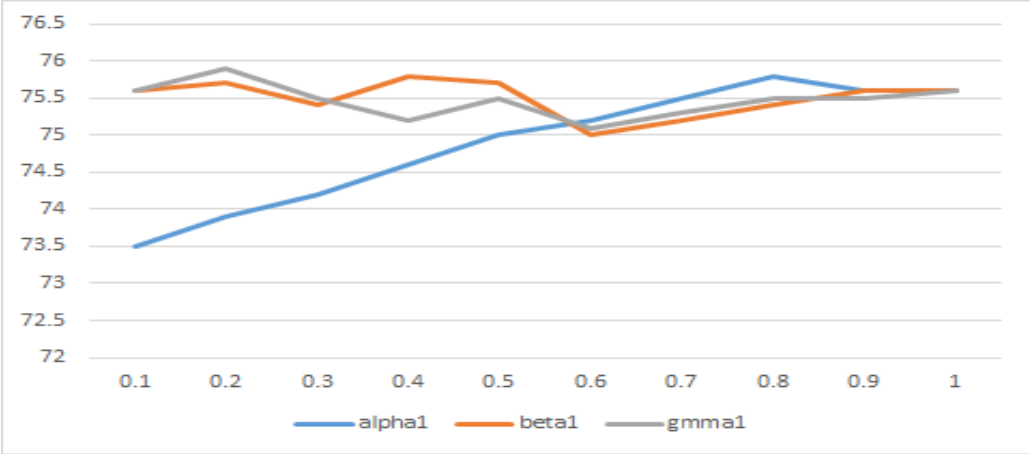


Figure 5.4: Evaluation of three parameters in formula 5.17

between text embedding of an e_{uu} link. $L_{tl}(u_i, u_j)$ is the loss between the text embedding of u_i and the link embedding of u_j. $L_{lt}(u_i, u_j)$ is the loss between the link embedding of a user u_i and text embedding of a user node u_j. $\alpha_1, \beta_1, \gamma_1$ are three weighted parameters for the three loss functions. When we evaluate the effect of α_1 in formula 5.17, we take the Bilibili dataset using 85% of nodes and links and the value of α_1 ranges from 0.1 to 1 with the increment of 0.1 in each step, while the other two parameters are fixed to 1. We use the same process to evaluate the effect of two other parameters β_1, γ_1 in function 5.17. The evaluation process of other parameters in function 5.21, and 5.22 are the same as Function 5.17.

Figure 5.4 shows the effect of three parameters in formula 5.17, we observe that α_1 affects the performance dramatically. Decreasing value of α_1 has negative impact. On the other hand, the change of β_1 and γ_1 will have much less effect on the performance of LTEH-A model. This is because the increasing value of α_1 essentially means increasing the significance of text embedding, which indicates that text embedding plays an important role in encoding user-to-user relationships.

Changing the weight of parameters in Formula 5.21, and Formula 5.22 has much less impact on the performance of LTEH-A model. We can still observe that the increasing of

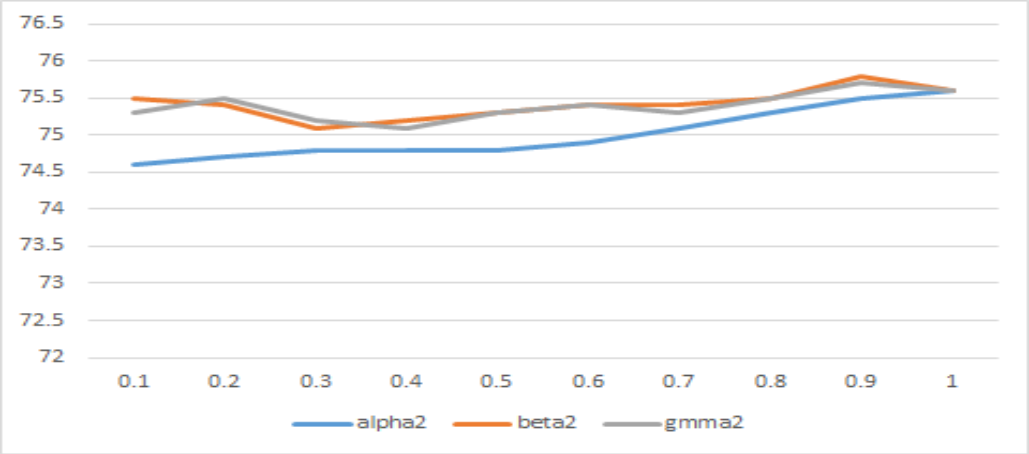


Figure 5.5: Evaluation of three parameters in formula 5.21

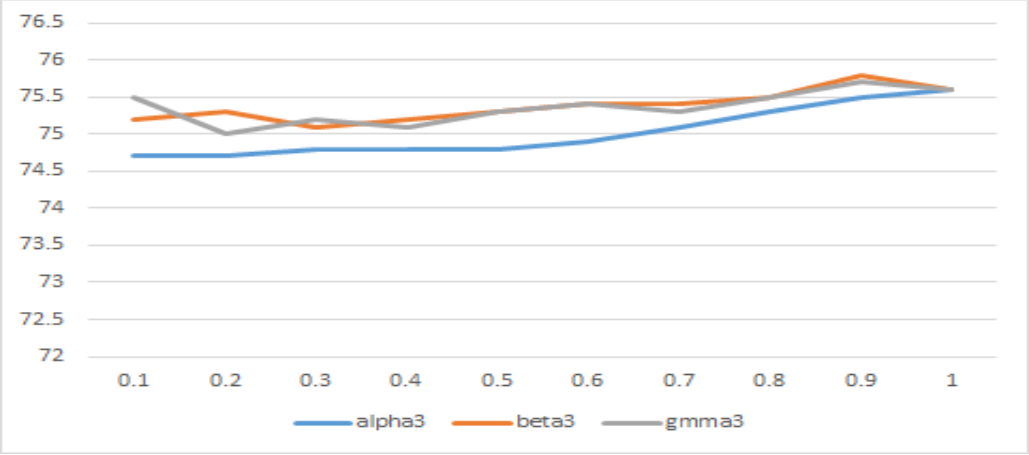


Figure 5.6: Evaluation of three parameters in formula 5.22

α_2, α_3 can improve the performance of LTEH-A, while β_2, β_3 and γ_2, γ_3 have relatively small effect on the performance of LTEH-A. The evaluations of the three parameters in the three functions indicate that text information of a link encodes valuable information in network embedding.

Evaluation of attention mechanism

To examine the effectiveness of the attention mechanism using link extensions in our proposed attention mechanism, we split our attention mechanism without changing any other

	Model	15%	25%	35%	45%	55%	65%	75%	85%	95%
Cora	Local-Text	82.1	83.5	84.4	86.6	90.2	91.8	92.0	93.1	94.4
	Text&Link	**83.5**	**86.5**	**90.4**	**86.5**	**92.3**	**92.9**	**93.3**	**94.5**	**95.4**
Hepth	Local-Text	85.1	86.9	87.6	87.9	88.8	90.2	91.7	93.4	95.5
	Text&Link	**87.9**	**88.5**	**88.9**	**90.7**	**90.8**	**92.9**	**93.4**	**96.1**	**96.8**
Zhihu	Local-Text	59.8	62.3	65.9	69.5	68.9	70.4	72.4	73.1	76.4
	Text&Link	**61.9**	**64.8**	**68.3**	**71.9**	**72.5**	**73.4**	**74.2**	**74.5**	**78.9**
Bilibili	Local-Text	58.8	59.3	61.4	64.5	68.9	69.0	70.7	74.1	76.4
	Text&Link	**60.1**	**62.1**	**64.2**	**66.2**	**68.5**	**69.9**	**71.5**	**75.6**	**78.0**

Table 5.9: AUC of Local-Text based attention mechanism vs. Text&Link based attention mechanism in LTEH

part of the algorithm by using (1) only local text as context for attention, labeled as **Local-Text** and (2) local text with extended context using both local text and extended text by links, labeled as **Text&Links** by our model.

Table 5.9 gives the performance of the two different attention mechanisms in link prediction task measured by AUC. The improvements in all the four datasets are clear and substantial. In Zhihu and Bilibili, text information is far richer than Cora and Hepth. For Zhihu, Text&Link achieves 2.50% net increase on average in AUC compared to Local-Text in different training proportions; in Bilibili, the average net increase is 1.73%. The average net increases reach 2.02% in Cora and 2.11% in Hepth. The result of this part proves that incorporating link information into attention mechanism can provide more information about node characteristics.

Visualization

Performance evaluation can also be observed by producing a visualization of a network in two-dimensional space. Visualizations can help to understand network topology. Visualization in our work is performed on the node representation of a 200 dimension node vector by the t-SNE algorithm [142][12] to reduce the dimensionality to 2.

[12] https://lvdmaaten.github.io/tsne/

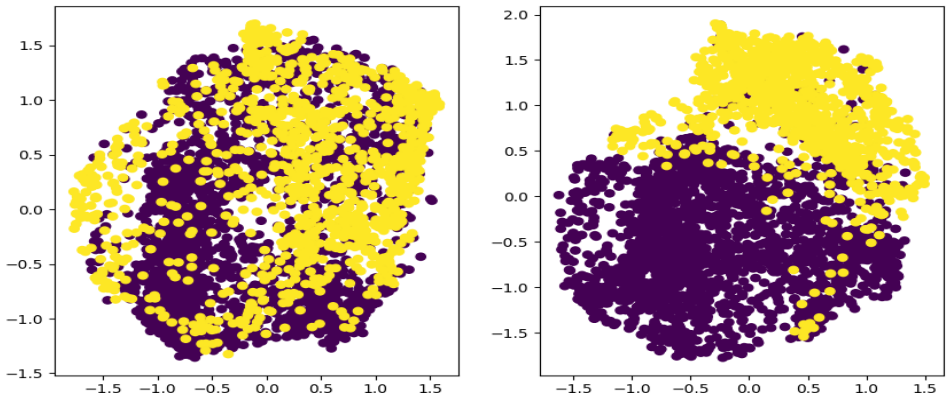


Figure 5.7: Visualization of two node types on Bilibili dataset(Left:CANE, Right:LTEH-A)

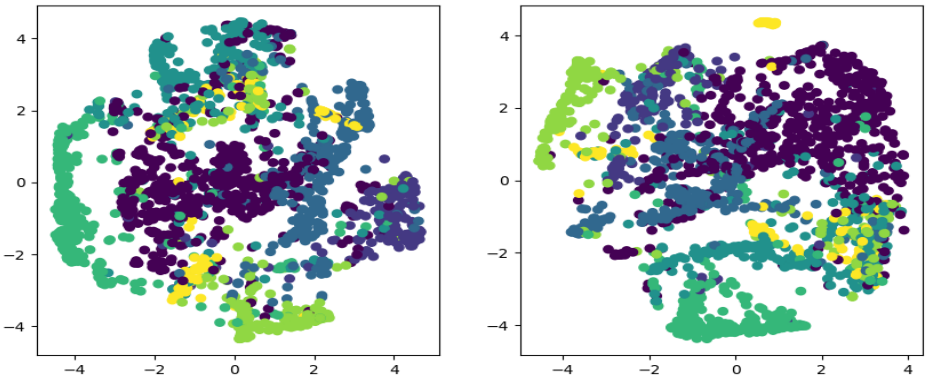


Figure 5.8: Visualization of seven types of user nodes in Cora dataset (Left:CANE, Right:LTEH-A)

Figure 5.7 shows the visualization of node types for the Bilibili dataset with the state-of-the-art system CANE-A on the left and our proposed LTEH-A on the right in the training ratio of 0.95. The yellow color represents the video nodes and the purple represents user nodes. It is easy to see that LTEH-A can separate the two types of nodes much better.

Figure 5.8 shows the visualization results of CANE-A and LTEH-A in Cora by reducing the dimensions to 7 groups. Even though Cora is a homogeneous network with only authors, each author belongs to one of 7 different categories of authors. Visualiza-

tion result shows that our model LTEH-A on the right still makes comparable result to the state-of-the-art CANE-A on the left.

5.3.3 Conclusion on learning user profiles from text and links

In this section, we present a novel model to learn node embedding for heterogeneous networks through a joint learning framework of both network links and text associated with nodes. The novelty of our proposed model includes two parts. Firstly, we learn the embedding of different nodes separately from links and other types of contents. Hence our model is capable of learning different types of nodes in heterogeneous networks. Secondly, we propose a novel attention mechanism to extend text by following links of adjacent nodes such that much larger context of the network can be included.

5.4 Chapter summary

This chapter present work to improve user profiling from two perspectives. Firstly, we propose a novel approach to predict user preferences by learning from both observed comments and missing comments based on the missing-not-at-random hypothesis. More specifically, we first make use of a user-word heterogeneous network embedding model to obtain both user and word representations in observed comments. We then construct a user-word matrix and a user-user similarity matrix to model missing comments by users. Both missing comments and observed comments are then consolidated to obtain the final user-to-user presentation through joint weighted matrix factorization to include missing comments in the final representation. This work indicate that missing comments does not follow the missing at random hypothesis and user inclination can still be learn even if they are mostly silent. To further leverage on social network links available in social media, we explore methods to extend the context of user profiles through network links. A novel method is proposed to learn networks node embedding in a network by using both link

and text information. This study proves that in a network with content information such as text, users, and other attributes, these content can help to distinguish different types of nodes. Embedding of different node types are separately processed, yet jointly optimized.

Currently, we have not considered more fine-grained relations between missing words and observed words. More fine grained relation can be obtained by dependency parser or external lexical resources. Future work can focus on fine grained relation to enrich user inclination information. For extending context, similar methods can be used in other semi-structured or unstructured content in social media data, such as images and animation videos, etc. This would be the future direction of our work.

Chapter 6

Incorporating user profiles into emotion analysis

By commonsense we know that SR text written by a person may be subjective or biased towards his/her own preferences. Review text can be written for commercial products such as cell phones, camera, or personal computers etc. It can also be reviews for movies, books, or sport matches. This chapter focuses on how user profiles can be better incorporated in emotion analysis for review text. As current work in review text mostly focuses on sentiment analysis, we focus on sentiment analysis in our investigation in this chapter. As we know, the Internet comments a person writes, especially review text, can influence emotion analysis results [173, 91]. Lenient users tend to give higher ratings than finicky ones even if they review the same products. On the other hand, popular products do receive higher ratings than those unpopular ones because the aggregation of user reviews still shows the difference in opinions for different products.

Including user profiles in learning models for sentiment analysis is not new. Recent tasks using neural networks models have already tried to incorporate user profile information together with product information in opinion analysis [67, 214, 33, 52]. User profiles (or product information) into product reviews are incorporated into different neural network models including CNN [67], RNN [214], LSTM [33], and Memory Network [52]. Among these models, the memory network model proposed by Dou's work [52]

is regarded as the newest state-of-the-art method. In Dou's proposed memory network, user profiles and product information are incorporated together as a single memory. This memory is built from an array of individually learned document representation so at to capture information at a much larger context. However, all previous works, including Dou's newest state-of-the-art model, handle user profile and product information in a unified model. User profiles and product information are not independent of each other in opinion analysis. User profiles are encoded in all the documents they write and product information are also encoded in all the comments written by users. Yet, putting such information together in a unified model may not be able to capture user profile information or product appropriately.

Even though user and product both play crucial roles in sentiment analysis, they are fundamentally different. The bellow example shows the difference between user and product:

Example 6.1 (Different background information influence the results)**.** *In a review about movie video v posted by user u, u said "The movie is so good and touching". From the perspective of this user, u maybe has a mean personality, even the review content is somehow positive, but u only give 2 stars out of 5. If the user u is a lenient, then he maybe gives all the movies or products 5 stars. From the perspective of this video, the topic of v may be easy to touch people and make people emotional, even most of the reviews about v is very positive, but maybe the actual quality is only 2 stars out of 5.*

Reviews written by a user can be affected by user profiles which are more subjective whereas reviews for a product are useful only if they are from a collection of different reviewers, because we know individual reviews can be biased. The popularity of a product tends to reflect the general impression of a collection of users as an aggregated result. Therefore, sentiment analysis of a product should give dual consideration to individual users as well as all reviews as a collection. To process user profile and product informa-

tion in a unified model may not be able to learn salient features of users and products effectively.

Based on the individual user profiles learned in Chapter 5, we propose to learn user profiles as a collection and product information as a collection using separate memory networks before making a joint prediction on sentiment classification. Firstly, we investigate how to use the memory network model to represent a collection of profiles by an array of individual user profiles. To capture a larger context of products, we can also build a memory of products as an array to include larger context of products. Once both user profile memory and product memory are learned, they can be incorporated to learn the joint representation for opinion analysis. We name our proposed model Dual User and Product Memory Network (DUPMN) model because we have two separately built memory networks: a user memory network (UMN) and a product memory network (PMN) based on document representation of user comments and product reviews.

To validate the effectiveness of our proposed model, evaluations are conducted on three benchmarking review datasets from IMDB and Yelp data challenge (including Yelp 13 and Yelp 14) [214]. Experimental results show that our algorithm can outperform baseline methods by large margins. Compared to the state-of-the-art method, DUPMN made 0.6%, 1.2%, and 0.9% increase in accuracy with *p-values* 0.007, 0.004, and 0.001 in the three benchmark datasets respectively. Results show that leveraging user profile and product information separately can be more effective for sentiment analysis.

The rest of this chapter is organized as follows. Section 6.1 gives related work, especially memory network models. Section 6.2 introduces our proposed DUPMN model. Section 6.3 gives the evaluation compared to state-of-the-art methods on three datasets. Section 6.4 concludes this chapter and gives some future directions in EA models to consider individual bias.

6.1 Related work

Recently, some state-of-the-art tasks use neural network based memory network (MemNN) model to construct an end-to-end learning model so that interaction between different elements can be better learned [239, 209, 6]. The Memory network model has been used in many NLP tasks like Question Answering [239] , Chatbot [6], and Sentiment analysis [216, 52]. A memory network model is composed of several inference components combined with a so called **memory** [239]. A memory is a matrix to contain a collection of information by a set of individual objects. The collection of information can either contain needed context [44], an external knowledge base [95], or a set of global users and products information [216]. The memory can represent a larger scale of information because it is built from a collection of objects instead of any individual object. [209].

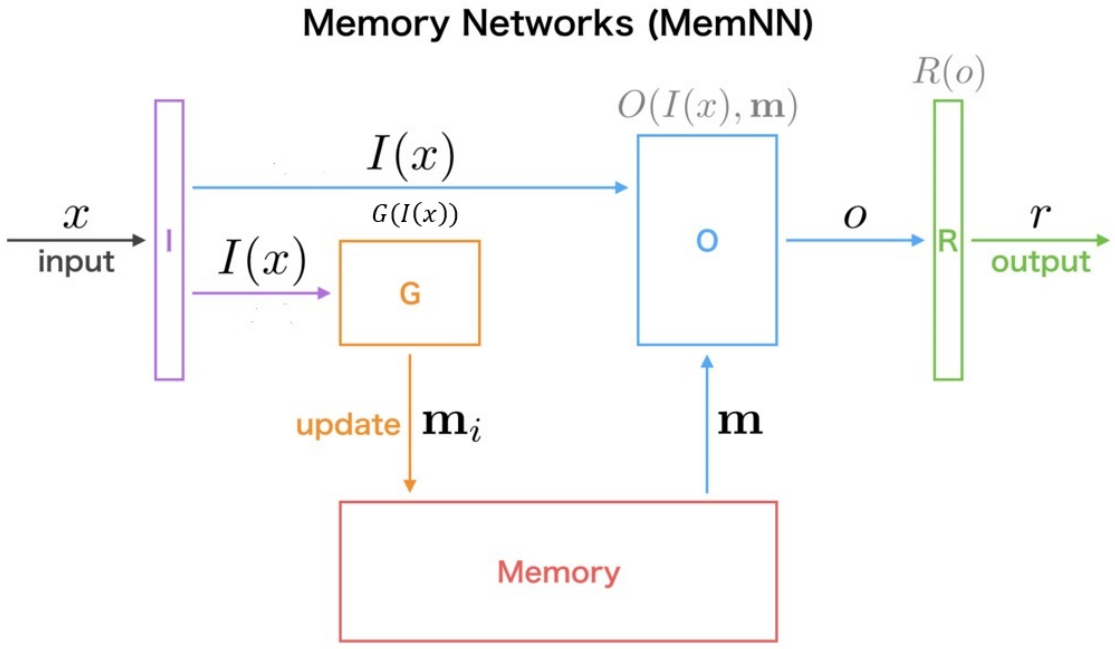


Figure 6.1: A one hop memory network model

A MemNN model can have either a single hop or multiple hops[209], similar to other neural network models. To make things simple, we show a single hop MemNN architec-

ture in Figure 6.1 as it was introduced by Weston et al. [239]. A MemNN model is built by a memory m (an array of objects) and four other major components. Given an input x, the **Input Feature Map Component**, denoted by I, first converts x to the needed internal feature representation I. x can be an input word, a sentence or a document depending on object of interest. Then, the **Generalization Component**, denoted by G, updates the old memories m_i given the new input. The simplest form of G only process with $I(x)$, $m = G(I(x))$, which is shown in Figure 6.1. More sophisticated variants of G can go back and update earlier stored memories or all memories based on the new evidence from the current input $I(x)$. The process is called generalization as there is an opportunity for the network to compress and generalize its memories at this stage for some intended future use. The **Output Feature Map Component**, denoted by O, produces a new output o (in the feature representation space), given the input and the updated m $o = O(I(x), m)$. The **Response Component**, denoted by R, decodes the output o to give the final response: $r = R(o)$. r can be a text response, an action, or a classification label.

In sentiment analysis, Tang et al. [216] propose a sentiment classification model utilizing memory networks which build a memory to hold local text information. As Tang's work did not consider user and product information, Dou [52] proposes a memory network utilizing user and product information for document sentiment classification. In Dou's model, user and product information compose the memory part to reflect the context in final rating prediction. However, since Dou's works in memory network model use single memory model to incorporate user profile and product information together, the unified model may not be able to learn salient features of users and products effectively.

6.2 User and product memory network model

In this chapter, we propose a DUPMN model. Firstly, document representation is learned by a hierarchical LSTM network to obtain both sentence level representation and doc-

ument level representation [211]. A memory network model is then trained using dual memory networks, one for training user profiles and the other for training product reviews. Both of them are joined together to predict sentiment for documents.

6.2.1 Task definition

Let D be the set of review documents for classification, U be the set of users, and P be the set of products. For each document $d(d \in D)$, user $u(u \in U)$ is the writer of d on product $p(p \in P)$. Let $U_u(d)$ be all documents posted by u and $P_p(d)$ be all documents on p. $U_u(d)$ and $P_p(d)$ define the user context and the product context of d, respectively. For simplicity, we use $U(d)$ and $P(d)$ directly. The goal of a sentiment analysis task is to predict the sentiment label for each d.

6.2.2 Document embedding

Since review documents for emotion classification such as restaurant reviews or movie comments are normally very long, a proper method to embed the documents is needed to speed up the training process and achieve better accuracy. Inspired by the work of Chen [33], a hierarchical LSTM network is used to obtain embedding representation of documents. The first LSTM layer is used to obtain sentence representation by the hidden state of an LSTM network. The same mechanism is also used for document-level representation with sentence level representation as input. User and product attentions are included in the network so that all salient features are included in document representation. For document d, its embedding is denoted as $\vec{d}$. $\vec{d}$ is a vector representation with dimension size n. In principle, the embedding representation of user context of d, denoted by $\hat{U}(d)$, and product context $\hat{P}(d)$ vary depending on d. For easy matrix calculation, we take m as our model parameter so that $\hat{U}(d)$ and $\hat{P}(d)$ are two fixed $n \times m$ matrices.

6.2.3 Memory network structure

Inspired by the successful use of memory networks in language modeling, question answering, and emotion analysis [209, 216, 52], we propose our DUPMN by extending a single memory network model to two memory networks to reflect different influences from users' perspective and products' perspective. The structure of the model is shown in Figure 6.2 with 3 hops as an example although in principle a memory network can have K computational hops.

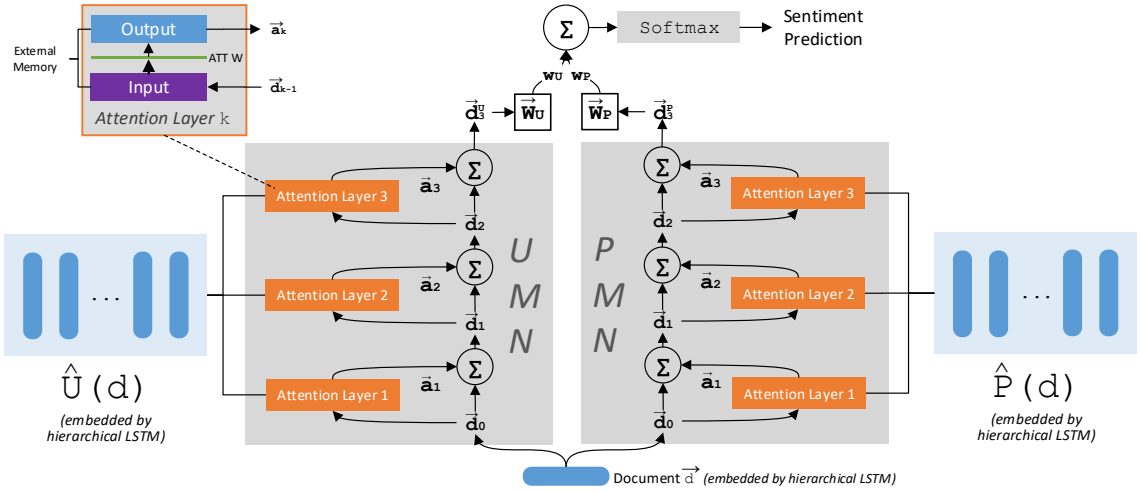


Figure 6.2: Structure for proposed DUPMN model

The DUPMN model has two separate memory networks: the UMN and the PMN. Each hop in a memory network includes an attention layer $Attention_i$ and a linear addition Σ_k. Since the external memory $\hat{U}(d)$ and $\hat{P}(d)$ have the same structure, we use a generic notation $\hat{M}$ to denote them in the following explanations. Each document vector $\vec{d}$ is fed into the first hop of the two networks ($\vec{d_0}$=$\vec{d}$). Each $\vec{d}_{k-1}$(k= 1 ..... K-1) passes through the attention layer using an attention mechanism defined by a softmax function to obtain the attention weights $\vec{p}_k$for document d:

$$\vec{p}_k = Softmax(\vec{d}_{k-1}^T * \hat{M}), \tag{6.1}$$

And to produce an attention weighted vector $\vec{a}_k$ by

$$\vec{a}_k = \sum_{i=0}^{m} p_{ki} * \vec{M}_i. \tag{6.2}$$

$\vec{a}_k$ is then linearly added to $\vec{d}_{k-1}$ to produce the output of this hop as $\vec{d}_k$.

After completing the Kth hop, the output $\vec{d}_K^u$ in UMN and $\vec{d}_K^p$ in PMN are joined together using a weighted mechanism to produce the output of DUPMN, $Output_{DUPMN}$, which is given below:

$$Output_{DUPMN} = w_U \vec{W}_U \vec{d}_K^u + w_P \vec{W}_P \vec{d}_K^p. \tag{6.3}$$

Two different weight vectors $\vec{W}_u$ and $\vec{W}_p$ in Formula 6.3 can be trained for UMN and PMN. w_U and w_P are two constant weights to reflect the relative importance of user profile $\vec{d}_K^u$ and product information $\vec{d}_K^p$. The parameters in the model include $\vec{W}_U$, $\vec{W}_P$, w_U and w_P. By minimizing the loss, those parameters can be optimized.

Final emotion label classification is obtained through a $Softmax$ layer. The loss function is defined by the cross entropy between the prediction from $Output_{DUPMN}$ and the ground truth labels.

6.3 Experiment and result analysis

Performance evaluations are conducted on three datasets and DUPMN is compared with a set of commonly used baseline methods including the state-of-the-art LSTM based method [33].

6.3.1 Datasets and evaluation matrix

The three benchmarking datasets include movie reviews from IMDB, restaurant reviews from Yelp 13 and Yelp14 developed by Tang [214]. All datasets are tokenized using the Stanford NLP tool [145]. Table 6.1 lists statistics of the datasets including the number of

	IMDB	Yelp13	Yelp14
#class	10	5	5
#doc	84,919	78,966	231,163
#users	1,310	1,631	4,818
#products	1,635	1,631	4,194
Av sen. len	24.56	17.37	17.25
Av docs/user	64.82	48.41	47.97
Av docs/prod	51.93	48.41	55.12
#p(0-50)	1,223	1,299	3,150
#p(50-100)	318	254	749
#p(100-150)	72	56	175
#p(150-200)	22	24	120

Table 6.1: Statistics of the three benchmark datasets

classes, number of documents, average length of sentences, average number of documents per user, and average number of documents per product. Since postings in social networks by both users and products follow the long tail distribution [109], we only show the distribution of total number of posts for different products. For example, #p(0-50) means the number of products which have reviews between the size of 0 to 50. In Figure 6.3. We can find that the distribution of data follows the long-tail distribution, most of the documents' numbers are within 1-100 per user or product. We split train/development/test sets at the rate of 8:1:1, following the same setting in [219, 33]. The best configuration by the development dataset is used for the test set to obtain the final result. [1]

Then the performance metrics that we used to measure the performance of the models are defined as following. In order to compare with baseline models, Accuracy, MAE and RMSE are used as measures for divergences between results of different models, as most of our selected baselines use this three evaluation matrices. They are defined as follows: T is number of correct predictions, N is the size of the testing set, and py_i and gy_i are prediction and ground truth for each training or testing record:

[1] N/A means the original paper did not provide the MAE value.

		IMDB			Yelp13			Yelp14		
	Model	Acc	RMSE	MAE	Acc	RMSE	MAE	Acc	RMSE	MAE
G1	Majority	0.196	2.495	1.838	0.392	1.097	0.779	0.411	1.060	0.744
	Trigram	0.399	1.783	1.147	0.577	0.804	0.487	0.569	0.814	0.513
	TextFeature	0.402	1.793	1.134	0.572	0.800	0.490	0.556	0.845	0.520
	AvgWordvec	0.304	1.985	1.361	0.530	0.893	0.562	0.526	0.898	0.568
G2	SSWE	0.312	1.973	N/A	0.549	0.849	N/A	0.557	0.851	N/A
	RNTN+RNN	0.400	1.734	N/A	0.574	0.804	N/A	0.582	0.821	N/A
	CLSTM	0.421	1.549	N/A	0.592	0.729	N/A	0.637	0.686	N/A
	LSTM+LA	0.443	1.465	N/A	0.627	0.701	N/A	0.637	0.686	N/A
	LSTM+CBA	0.489	1.365	N/A	0.638	0.697	N/A	0.641	0.678	N/A
G3	UPNN	0.435	1.602	0.979	0.608	0.764	0.447	0.596	0.784	0.464
	UPDMN	0.465	1.351	0.853	0.613	0.720	0.425	0.639	0.662	0.369
	InterSub	0.476	1.392	N/A	0.623	0.714	N/A	0.635	0.690	N/A
	LSTM+UPA	0.533	1.281	N/A	0.650	0.692	N/A	0.667	0.654	N/A
New	DUPMN	**0.539**	**1.279**	**0.734**	**0.662**	**0.667**	**0.375**	**0.676**	**0.639**	**0.351**

Table 6.2: Evaluation of different methods; best result/group is marked bold; second best is underlined.

$$Accuracy = \frac{T}{N} \tag{6.4}$$

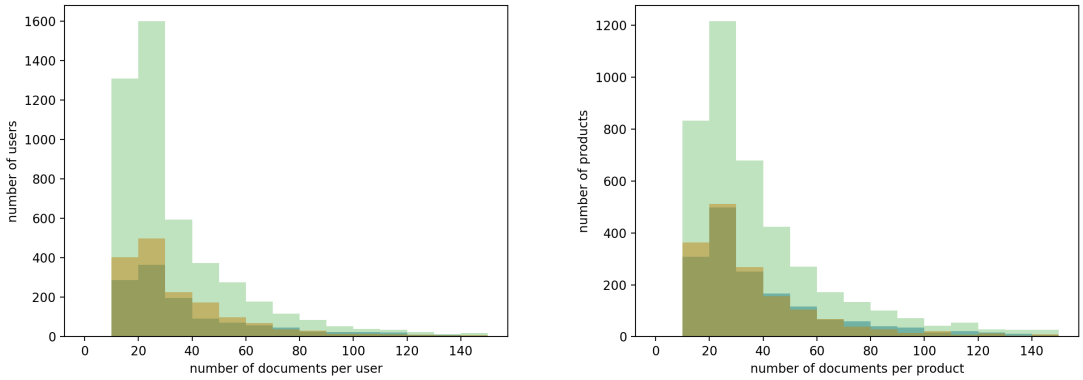
$$MAE = \frac{\sum_i |py_i - gy_i|}{N} \tag{6.5}$$

$$RMSE = \sqrt{\frac{\sum_i (py_i - gy_i)^2}{N}} \tag{6.6}$$

6.3.2 Baseline methods

In order to make a systematic comparison, three groups of baselines are used in the evaluation. The first group of methods are simple baseline methods using commonly used linguistic features including:

- **Majority**: A simple majority classifier based on sentence labels.



(a) Statistic of documents # per user (b) Statistic of documents # per product

Figure 6.3: Number of documents per user/product for three datasets

- **Trigram**: An SVM classifier using unigram/bigram/trigram as features.
- **Text feature**: An SVM classifier using word level and context level features, such as n-gram and sentiment lexicons.
- **AvgWordvec**: An SVM classifier that takes the average of word embeddings in Word2Vec as document embedding.

All feature sets except Majority use the SVM classifier.

The second group of methods includes recent sentiment classification algorithms which are top performers for review text including state−of−the−art models not using user or product information. Below gives the list of Group 2 methods:

- **SSWE** [218] — An SVM model using sentiment specific word embedding.
- **RNTN+RNN** [203] — A Recursive Neural Tensor Network (RNTN) to represent sentences and trained using RNN.
- **CLSTM** [245] — A Cached LSTM model to capture overall semantic information in long text.

- **LSTM+LA** [33] — A state-of-the-art LSTM using local context as attention mechanism in both sentence level and document level.

- **LSTM+CGA** [134]— A state-of-the-art LSTM model using cognition based data to build attention mechanism.

The third group includes the recent state-of-the-art models using both user and product information. Group 3 methods include:

- **UPNN** [215] — User and product information for sentiment classification at document level based on a CNN network.

- **UPDMN** [52] — A memory network for document-level sentiment classification by including user and product information by a unified model. Hop 1 gives the best result, and thus K=1 is used.

- **InterSub** [67] — A CNN model making use of network embedding of user and product information.

- **LSTM+UPA** [33] — the state-of-the-art LSTM including both local context based attentions and user/product in the attention mechanism at both sentence level and document level.

For the DUPMN model, we also include two variations which use only one memory network: The first variation only includes user profiles in the memory network, denoted as **DUPMN-U**. The second variation only uses product information, denoted as **DUPMN-P**.

6.3.3 Experimental results and discussion

Four sets of experiments are conducted. The first experiment compares DUPMN with other sentiment analysis methods. The second experiment evaluates the effectiveness of different hop size K of memory network. The third experiment evaluate the effectiveness

of UMN and PMN in different datasets. The fourth set of experiment examines the effect of memory size m to the performance of DUPMN. Performance measures include Accuracy(ACC), Root-Mean-Square-Error (RMSE), and Mean Absolute Error (MAE) for our model. For other baseline methods in Group 2 and Group 3, their reported results are used. We also show the p-value by comparing the result of 10 random tests for both our model and the state-of-the-art model [2] in t-test [3].

	IMDB			Yelp13			Yelp14		
Model	Acc	RMSE	MAE	Acc	RMSE	MAE	Acc	RMSE	MAE
Majority	0.196	2.495	1.838	0.392	1.097	0.779	0.411	1.06	0.744
Trigram	0.399	1.783	1.147	0.577	0.804	0.487	0.569	0.814	0.513
TextFeature	0.402	1.793	1.134	0.572	0.800	0.490	0.556	0.845	0.520
AvgWordvec	0.304	1.985	1.361	0.530	0.893	0.562	0.526	0.898	0.568
SSWE	0.312	1.973	N/A	0.549	0.849	N/A	0.557	0.851	N/A
RNTN+RNN	0.400	1.734	N/A	0.574	0.804	N/A	0.582	0.821	N/A
CLSTM	0.421	1.549	N/A	0.592	0.729	N/A	0.637	0.686	N/A
LSTM+LA	0.443	1.465	N/A	0.627	0.701	N/A	0.637	0.686	N/A
LSTM+CBA	0.489	1.365	N/A	0.638	0.697	N/A	0.641	0.678	N/A
UPNN(K)	0.435	1.602	0.979	0.608	0.764	0.447	0.596	0.784	0.464
UPDMN(K)	0.465	1.351	0.853	0.613	0.720	0.425	0.639	0.662	0.369
InterSub	0.476	1.392	N/A	0.623	0.714	N/A	0.635	0.690	N/A
LSTM+UPA	0.533	1.281	N/A	0.650	0.692	N/A	0.667	0.654	N/A
DUPMN	**0.539**	**1.279**	**0.734**	**0.662**	**0.667**	**0.375**	**0.676**	**0.639**	**0.351**

Table 6.3: Experimental results of DUPMN and comparison models[4]

Table 6.2 shows the result of the first experiment. DUPMN uses one hop (the best performer) with m being set at 100, a commonly used dimension size for memory networks. Generally speaking, Group 2 performs better than Group 1. This is because Group 1 uses a traditional SVM with feature engineering [29] and Group 2 uses more advanced deep learning methods proven to be effective by recent studies [105, 33]. However, some

[2] We re-run experiment based on their public available code on github (https://github.com/thunlp/NSC.).

[3] http://www.statisticshowto.com/probability-and-statistics/t-test/

[4] Best results are marked in bold; second best are underlined in the table

feature engineering methods are no worse than some deep learning methods. For example, the TextFeature model outperforms SSWE by a significant margin.

When comparing Group 2 and Group 3 methods, we can see that user profiles and product information can improve performance as most of the methods in Group 3 perform better than methods in Group 2. This is more obvious in the IMDB dataset which naturally contains more subjectivity. In the IMDB dataset, almost all models with user and product information outperform the text-only models in Group 2 except LSTM+CBA [134]. However, the two LSTM models in Group 2 which include local attention mechanism do show that attention base methods can outperform methods using user profile and product information. In fact, the LSTM+CBA model using attention mechanism based on cognition grounded eye tracking data in Group 2 outperforms quite a number of methods in Group 3. LSTM+CBA in Group 2 is only inferior to LSTM+UPA in Group 3 because of the additional user profile and production information used in LSTM+UPA.

Most importantly, DUPMN model with both user memory and product memory significantly outperforms all the baseline methods including the state-of-the-art LSTM+UPA model [33]. By using user profiles and product information in memory networks, DUPMN outperforms LSTM+UPA in all three datasets. In the IMDB dataset, our model makes 0.6 % improvement over LSTM+UPA in Accuracy with $p - value$ of 0.007. Our model also achieves lower RMSE value. In the Yelp review dataset, the improvement is even more significant. DUPMN achieves 1.2% improvement in accuracy in Yelp13 with $p - value$ of 0.004 and 0.9% in Yelp14 with $p - value$ of 0.001, the lower RMSE obtained by DUPMN also indicates that the proposed model can predict review ratings more accurately.

The second set of experiments evaluates the effectiveness of DUPMN using different number of hops K. Table 6.4 shows the evaluation results. The number in the brackets after each model name indicates the number of hops used. Two conclusions can be obtained from Table 6.4. We find that more hops do not bring benefit. In all the three models, the single hop model obtains the best performance. Unlike video and image in-

formation, written text is grammatically structured and contains abstract information such that multiple hops may introduce more information distortion. Another reason may be due to over-fitting by the additional hops.

Comparing the performance of DUPMN-U and DUPMN-P in Table 6.4, it also shows that user memory and product memory indeed provide different kinds of information and thus their usefulness is different in different datasets. For the movie review dataset, IMDB, which is more subjective, results show that user profile information uses DUPMN-U as there is a 1.3% gain compared to that of DUPMN-P. However, on restaurant reviews in Yelp datasets, DUPMN-P performs better than DUPMN-U indicating product information is more valuable.

To further examine the effects of UMN and PMN to sentiment classification, we observe the difference of optimized values of the constant weights w_U and w_P between the UMN and the PMN given in Formula 6.3. The difference in their values indicates the relative importance of the two networks. The optimized weights given in Table 6.5 on the three datasets show that user profile has higher weight than product information in IMDB because movie review is more related to personal preferences whereas product information has higher weight in the two restaurant review datasets. This result is consistent with the evaluation in Table 6.4 on DUPMN-U and DUPMN-P.

Figure 6.4 shows the change of w_U and w_P in a learning process of DUPMN for IMDB dataset. Table 6.5 shows the average combining weight w_U and w_P for all three benchmark datasets.

The figures of three data-sets show two different trends. Figure 6.4a shows in movie review, the weight of user goes up with the weight of product goes down, and the optimized weight shows user profile have higher weight than product information. Figures 6.4b and 6.4c show a different trend, while the product information has higher weight.

The result of the fourth set of experiments is shown in Figure 6.5 and Table 6.6. We

[5] Best results are marked in bold; second best are underlined in the table

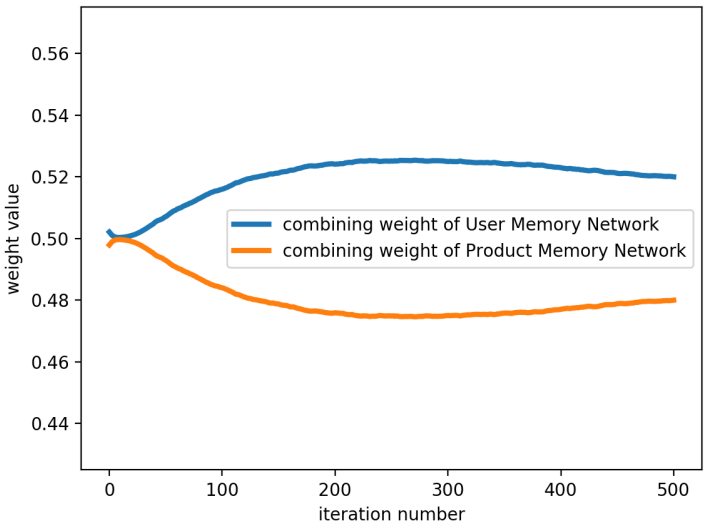
	IMDB			Yelp13			Yelp14		
	Acc	RMSE	MAE	Acc	RMSE	MAE	Acc	RMSE	MAE
DUPMN-U(1)	0.536	1.273	0.737	0.656	0.687	0.380	0.667	0.655	0.361
DUPMN-U(2)	0.526	1.285	0.748	0.653	0.689	0.382	0.665	0.661	0.369
DUPMN-U(3)	0.524	1.295	0.754	0.651	0.692	0.388	0.661	0.667	0.374
DUPMN-P(1)	0.523	1.346	0.769	0.660	0.668	0.370	0.670	0.649	0.357
DUPMN-P(2)	0.517	1.348	0.775	0.656	0.680	0.380	0.667	0.656	0.364
DUPMN-P(3)	0.512	1.356	0.661	0.651	0.699	0.388	0.661	0.661	0.370
DUPMN(1)	**0.539**	**1.279**	**0.734**	**0.662**	**0.667**	**0.375**	**0.676**	**0.639**	**0.351**
DUPMN(2)	0.522	1.299	0.758	0.650	0.700	0.390	0.667	0.650	0.359
DUPMN(3)	0.502	1.431	0.830	0.653	0.686	0.382	0.658	0.668	0.371

Table 6.4: Evaluation of different memory network hops and user and product information utilization[5]

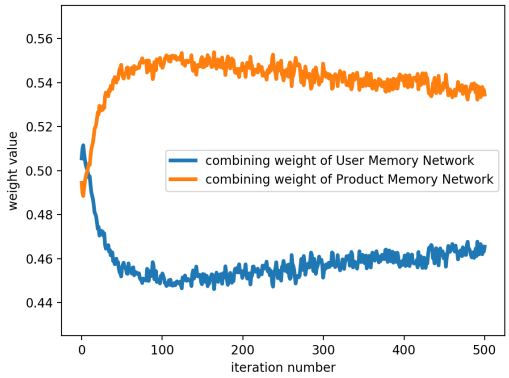
IMDB		Yelp13		Yelp14	
w_U	w_P	w_U	w_P	w_U	w_P
0.534	0.466	0.475	**0.525**	0.436	**0.564**

Table 6.5: Average combine weight

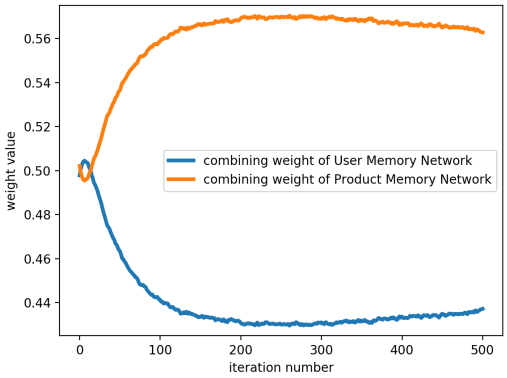
observed in the previous chapter that the most social network data follows long tail distribution. If the dimension size to represent the data is too small, some context information will be lost. On the other hand, too large a dimension size which requires more resources in computation and storage may not introduce much benefit. Thus, the fourth set of experiments evaluates the effect of dimension size m in the DUPMN memory networks. Figure 6.5 shows the result of the evaluation for 1 hop configuration with memory size starting at 1 with 10 points at each increment until size of 75 and 25 point increment from 75 to 200 to cover most postings. Results show that when memory size increases from 10 to 100, the performance of DUPMN steadily increases. Once it goes beyond 100, DUPMN is no longer sensitive to memory size. This is related to the distribution of document frequency rated by user/product in Table 6.1 as the average is around 50 or so. With long tail distribution, after 75, not many new documents will be included in the context. To improve algorithm efficiency without much compromise on performance, m can be any



(a) for IMDB dataset



(b) for Yelp13 dataset



(c) for Yelp14 dataset

Figure 6.4: The change of w_U and w_P in a learning process of DUPMN for datasets

value that doubles the average. So, values between 100-200 in our algorithm should be quite sufficient.

6.3.4 Feature analysis

This experiment examines features extracted for users as compared to that of products. Feature analysis is conducted in two parts. The first part shows the difference in features

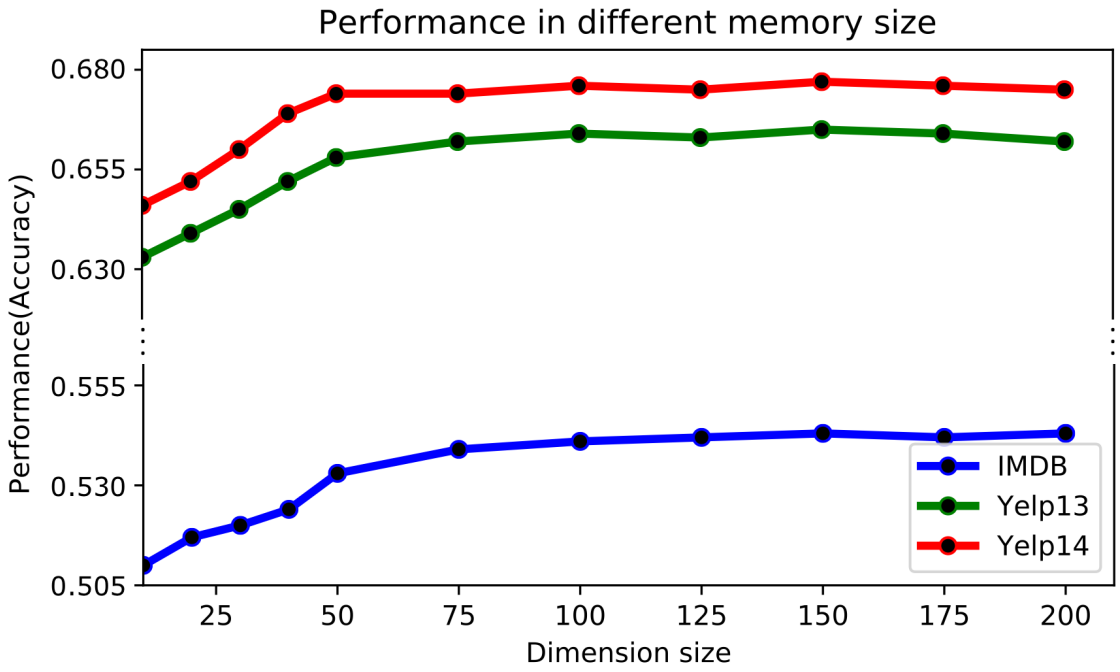


Figure 6.5: Effect of different memory sizes

extracted by user memory and product memory. The second part examines the use of adjectives in the two memories.

Figure 6.6 [6] shows two groups of word cloud graphs for IMDB dataset. The two upper sub-figures in Figure 6.6 shows two word cloud graphs that demonstrate the word frequency of reviews of the top 10 users giving highest ratings (lenient raters) and 10 users who give average lowest ratings(finicky raters) to movies in IMDB. Note that the high-frequency words include both personal feelings and product description but using different polarities. Personal feelings include words such as *like* (positive), *bad* (negative), etc. and movie description words include: *wonderful* (positive), *not great* (negative), etc. By contrast, words used in reviews for 10 highest or lowest rated movies, as shown in the two sub-figures in the bottom of Figure 6.6, are more objective, such as *old*, *new*, *little*, etc. Those words are mainly about the movies themselves rather than personal feelings.

The two restaurant review datasets show different characters. In the two upper sub-

[6] Word cloud tool is from (https://www.wordclouds.com/).

Memory Size	IMDB			Yelp13			Yelp14		
	Acc	RMSE	MAE	Acc	RMSE	MAE	Acc	RMSE	MAE
10	0.516	1.378	0.795	0.630	0.729	0.416	0.654	0.673	0.377
20	0.503	1.550	0.866	0.604	0.778	0.456	0.651	0.684	0.384
30	0.516	1.383	0.791	0.643	0.707	0.397	0.668	0.661	0.362
40	0.524	1.367	0.778	0.647	0.695	0.390	0.674	0.641	0.351
50	0.528	1.368	0.769	0.654	0.680	0.379	0.671	0.653	0.356
75	0.529	1.339	0.768	0.655	0.690	0.384	0.674	0.653	0.354
100	0.539	1.279	0.734	0.662	0.667	0.375	0.676	0.639	0.351

Table 6.6: Evaluation of different memory size

figures in Figure 6.7, it is hard to distinguish the best and worst raters. Even the worst raters use positive words like *better, great, fresh,* etc in a high frequency. But the product information, which reflects the popularity of the target restaurant in the lower two sub-figures Figure 6.7, shows a huge difference between the highest rating products and the lowest rating products. That can partly explain why product memory works better than user memory in the restaurant review datasets.

The second aspect of feature analysis shows the highest 20 adjectives for 10 users giving the highest ratings (lenient raters) and lowest ratings (finicky raters) as well as 10 highest rated products and 10 lowest rated products. Despite the difference between user profiles and product information, we observed that the huge gap between lenient user and finicky user. Table 6.7 and 6.8 show that in IMDB and Yelp 13, all the 20 highest adjective for lenient users are positive words, while the most of top 20 adjectives in finicky user are negative words. From product perspective, the top 20 adjectives for highest rating products are also all positive, while most frequent adjectives for lowest rating products are negative or positive words co-occur with negation (e.g: not). That indicates user profile and product information can provide information to sentiment analysis model. In the movie review dataset, the user profile are more effective in identify sentiment than product information, and the restaurant review shows different trend.

IMDB USER				IMDB PRODUCT			
HIGHEST		LOWEST		HIGHEST		LOWEST	
word	frequency	word	frequency	word	frequency	word	frequency
great	413	dislike/hate	566	great	531	(not) great	104
good	145	good	236	best	460	like	95
best	143	bad	228	like	458	good	86
excellent	95	great	138	most	390	best	78
wonderful	94	better	125	good	339	little	58
classic	93	original	110	wonderful	223	different	41
fantastic	85	big	109	greatest	185	delicious	39
funny	72	real	109	classic	164	amazing	34
brilliant	63	old	107	new	156	nice	29
dead	60	new	103	old	150	better	29
old	58	best	89	little	148	fresh	29
real	55	least	88	perfect	143	sweet	28
dark	54	few	87	better	135	perfect	28
little	53	funny	87	same	122	wonderful	27
like	52	dead	86	real	121	beautiful	26
better	52	stupid	69	another	118	before	26
original	52	boring	65	few	115	favorite	25
beautiful	45	black	63	silent	113	small	25
young	45	long	60	big	112	first	24
hilarious	44	salty	57	young	99	most	24

Table 6.7: Adjective frequency table of users and products with 10 highest and 10 lowest ratings in IMDB

Figure 6.6: Word clouds of reviews for 10 users who give average highest or lowest ratings (above), and 10 products which have average highest or lowest ratings (below) in IMDB

YELP13 USER				YELP13 PRODUCT			
HIGHEST		LOWEST		HIGHEST		LOWEST	
word	frequency	word	frequency	word	frequency	word	frequency
good	146	good	143	great	104	worst	128
great	135	great	97	like	95	(not) great	44
like	90	more	76	good	86	bad	34
best	77	better	58	best	78	nice	22
wonderful	56	fresh	51	little	58	little	21
fresh	44	before	50	different	41	different	21
delicious	37	hot	43	delicious	39	wrong	20
little	34	small	42	amazing	34	long	19
nice	33	little	41	nice	29	friendly	17
amazing	32	old	41	better	29	full	17
happy	32	green	33	fresh	29	free	17
tasty	32	bad	32	sweet	28	old	16
excellent	31	real	26	any	28	hard	15
first	30	nice	26	perfect	28	clean	14
favorite	30	new	24	wonderful	27	big	14
brilliant	26	high	22	beautiful	26	large	13
few	25	large	22	favorite	25	busy	13
friendly	23	horrible	21	new	25	extra	12
full	22	happy	20	small	25	expensive	12
hot	22	special	19	few	22	wrong	12

Table 6.8: Adjective frequency table of users and products with 10 highest and 10 lowest ratings in YELP 13

6.3.5 Case analysis

In this section, two example cases are analyzed to show the performance of DUPMN compared to LSTM+LA as case studies.

Below is the written text of User ID: 1150186 for a movie review from the IMDB dataset. The rating of 10 by the user, serves as the gold answer. The text is used by our both DUPMN and LSTM+LA.

> **Case 1 (User ID 1150186):** *okay , there are two types of movie lovers : the ones who watch one movie every six months and talk about it for the rest of the year, and the ones who actually watch movies all the time. people who belong to the first category*

Figure 6.7: Word clouds of reviews for 10 users who give average highest or lowest ratings (above), and 10 products which have average highest or lowest ratings (below) in Yelp13

, expect everything from a movie, let 's say, they expect to see a ' titanic ' every time they go to the cinema. the rest eventually learn to appreciate the good elements of a film , since they know how rare it is to find ' the perfect movie ' ." this movie sucks " ? well, I beg to differ. i mean, it is definitely better than other sci-fi films like 'armageddon' or even 'the phantom menace' no jar-jar here. The audio and visual effects are simply terrific and travolta's performance is brilliant - funny and interesting . What people expect from sci-fi movies is beyond me . When 'starship troopers' was released , absolutely the best space sci-fi movie of the 90 's, everyone said it was a bomb. Fortunately , it starts to gain some recognition over the last years , since the release of the dvd. same here, only worse. Why doesn't anyone care to mention the breath-taking effects or the captivating atmosphere ? what did they expect, a 6 movie saga to satisfy their hunger for sci-fi? At the time these lines are written, the imdb rating for 'battlefield earth' is below 2.5 , which is unacceptable for a movie with such craftsmanship. 'scary movie', possibly the worst movie of all time - including home made movies, has a 6 ! Maybe we should all be a little more subtle when we criticize movies like this and especially sci-fi movies, since they have

become an endangered genre. Have you seen any of the major studios produce sci-fi movies lately? Give this movie the recognition it deserves.

Author Rating: 10/10(serves as the gold answer,

LSTM+LA predicted rating: 1/10,

DUPMN predicted rating: 10/10.

In Case 1, the LSTM+LA method can only make prediction based on the text comment provided above. From the perspective of sequential context in this case, two reasons are likely to be behind the poor prediction result of LSTM+LA. Firstly, Case 1 has many negative words such as *unacceptable*, *criticize* and *sucks*. Secondly, in the first half of the comment, the author expressed strongly negative opinions. However, those negative opinions are towards certain movie lovers who actually not watch movies all the time. This large portion of text is likely to mislead the LSTM+LA model with the wrong prediction, although the author has a positive opinion of this movie as concluding remarks.

By contrast, the prediction of DUPMN is not purely based on the current piece of comment as it also learns from users' information from all his past comments which provide a much larger context. In the training data, the user, with ID 1150186 has 24 comments about other movies about 16 of them are sci-fi movies. Among all the sci-fi movies he gives 9 or 10 to all sci-fi movies in his rating record. Such information is indeed learned by our DUPMN to make a correct prediction based on training data. To show how larger context is included, below is another example comment written by the user with ID 1150186.

An example of comment posted by User 1150186: *I didn't even hear about this movie until a couple of months ago when i really got into science-fiction movies. I am glad i was able to pick up a copy for less than eight bucks because it was a huge bargain for a great movie . dark city follows a guy who wakes up in bathtub and finds a dead body in the next room. He then discovers he is wanted for a bunch of*

other murders . The catch is he doesn't remember a thing . I don't wanna say to much because you should really see it for yourself. who i had never heard of kiefer sutherland, jennifer connelly yummy and william hurt all give great performances the best though is from richard o'brien who plays the enigmatic and downright cool mr. hand , but the i think the real stars are the writers who make an original plot with plenty of clever twists and dialogue , and the set and graphic designers who make one of the coolest looking cities i have ever seen. if you love sci-fi and have never seen this movie before like me, i think you should check it out.

Author Rating: 9/10 (In training data).

Obviously, this user is a sci-fi movie lover shown by the text he wrote at the end of comment. Similar declaration appears in other comments made by this user. Based on the user memory network, DUPMN model is able to include these information in a larger context to correctly predicts the score contrasted to the failure of LSTM+LA.

However, the DMPMN model would work if a user can provide sufficient larger context. For users with a few additional comments to extend the context, DMPMN has no advantage over the LSTM+LA model. In the below case, because user 24733533 did not have many posted comments. For example, in the bellowing case, the user (User ID:24733533) has only one comment on training data.

Case 2 (User ID:24733533): *The sixth sense is the story about a young boy who lives with his mom that has troubles in school , and troubles interacting with all of the spoiled rich kids that he has in his school. He talks with a psychologist played by bruce will is. The psychologist tries to figure out what kind of problems the kid has, but then it turns out that the kid sees dead people. The movie isn't scary like everyone says, but it is very intriguing movie as you see what the kid sees, you see why he 's so horrified , and you see how he handles his fears . The ending is probably*

what the film is known for don't worry , i won't go anywhere near spoiling it for you . you would probably never guess it , but unfortunately, some jerk spoiled it for me . hopefully no one ever does the same to you . this is a very original movie. in some ways, i think that might Shyamalan was inspired by the creepy movie pet Sematary . if you watch both films, you might also notice a similarity . watch the sixth sense at all costs , it 's worth it.

User ID: 24733533,

Author rating:8 (serves as the gold answer),

LSTM+LA predicted rating:7,

DUMPN predicted rating: 6.

In case 2, the user has one comment on the training data. This far from sufficient to build a user profile. Therefore, both LSTM+LA and DUMPN must reply on text given in this comment to do prediction. Note that in Case 1, the first three sentences introduce the movie without subjective opinion. The user only starts to show his attitude in the fourth sentence, claiming that the movie is not scary but very intriguing. The emotion in the rest part is ambiguous until the author gives a thumb up to this movie in the last sentence. From word perspective, the comment contains both positive words such as "very intriguing", "very original", and "worth") and negative words such as "scary", "horrified", and "creepy". The comment in this case have a very complicated structure, which makes emotion prediction very difficult for both models.

6.4 Chapter summary

In this chapter, we present our proposed deep learning method using dual memory network model to make better use of user profiles and product information into sentiment analysis

for review text. We argue that user profile and product information are fundamentally different as user profiles contain more subjectivity whereas product reviews as a collection contain more salient features of products at the aggregated level.

Based on this hypothesis, two separate memory networks for user context and product context are built at the document-level through a hierarchical learning model. The inclusion of an attention mechanism can capture semantic information more effectively. Learning results from the dual memory networks serve as input to a unified classification model for optimization. Evaluation on three benchmark review datasets shows that our proposed DUPMN model outperforms the current state-of-the-art systems with significant improvements with the p-value of 0.007, 0.004, and 0.001, respectively. We also show that single hop is the most effective setting in the memory network model. Analysis on the contribution of user profile and product information demonstrates that they do have different performance effects on different datasets. In more subjective datasets such as IMDB, the inclusion of user profile information is more important. On the other hand, for more objective datasets such as Yelp data, collective information about restaurant reviews play a more important role in the classification.

Future work includes two directions. One direction is to explore the contribution of user profiles and product information in sentiment analysis tasks at the aspect level. Another direction is to explore how knowledge base can be incorporated to further improve the performance of sentiment classification.

Chapter 7

Conclusions and suggestions for future research

In this chapter, the main contributions of this thesis will be summarized first, followed by a discussion on limitations and future works.

7.1 Summary of Contributions of this Thesis

The main contributions of this thesis are summarized as follows:

1. **Linguistically driven model for emotion analysis (Chapter 3)**

   We approach emotion analysis from a novel perspective by incorporating linguistic features associated with orthography for social media text including the consideration of shifts of symbols between language scripts and the use of stylistic variations such as unconventional use of punctuation marks. The model is evaluated by three different types of datasets to test the hypotheses as a cross-domain dataset comparison. Results show that orthographic features are indeed linked to emotion classification in social media text although they play much less role to formal text. On the other hand, morpho-syntactic features contribute more to emotion classification in formal style text.

2. **Cognition grounded model for emotion analysis (Chapter 4)**

We explore the use of cognition grounded eye-tracking data to train attention models to improve the performance of linguistics-driven models. We build a novel cognition grounded attention (CGA) model for emotion analysis learned from cognition grounded eye-tracking data. This is one of the first attempt to use cognition grounded data to build attention models in emotion analysis. Evaluation on several real-world review datasets shows that our method outperforms the state-of-the-art methods significantly. Our work also indicates that both the quality and scale of eye-tracking data have great influence on the effectiveness of the cognition grounded attention model. We prove that cognition grounded data can be used to improve attention mechanisms and thus indirectly improves the performance of emotion analysis.

3. **User profile construction (Chapter 5)**

   We explore a novel approach to predict user preferences by learning from both observed comments and missing comments based on the missing-not-at-random hypothesis. To further leverage on social network links available in SR text, we explore methods to extend the context of user profiles through network links. We propose a novel approach to learn node embedding through a joint learning framework of both network links and text associated with nodes. The method can handle both homogeneous networks and heterogeneous networks with multiple types of links. Comparison to the state-of-the-art models using a number of datasets clearly indicates the advantage of our proposed method. Our work indicates that missing comments does not follow the missing at random hypothesis and user inclination can still be learn even if they are mostly silent. We also prove that in a user network learn from different content such as text, users, and other attributes can help to distinguish the profile of different types of nodes.

4. **Incorporating user profiles into emotion analysis (Chapter 6)**

We propose a deep learning method to make better use of biased user profile into emotion analysis for review text. The newly proposed machine learning model based on memory network framework using dual user and product memory networks. User profiles as a collection are aggregated by a memory network to encode user biases. A separate memory network is also used to learn product information. This is the first attempt to use dual memory networks to learn user profile and product information. Evaluation on three benchmark review datasets shows that the proposed DUPMN model outperforms the current state-of-the-art systems with significant improvements with p-values of 0.007, 0.004, and 0.001 respectively. Evaluation result shows that user profile and product information are indeed different and have different effect on different datasets. This framework can also be used to incorporate other biased information in emotion analysis.

7.2 Limitations and future work

Emotion analysis is a relatively new field, and it has attracted a lot of interest in recent years. As with most doctoral studies, the research presented here has many remaining questions. The limitations and future directions can be subjected in three aspects: resources, models, and feature selection.

Firstly, from resource perspective, the eye tracking data we used in our attention model are from a different domain, and the scale is very limited. We anticipate even greater improvement with a larger scale eye-tracking data in similar genre as the emotion analysis text. Making available more eye-tracking corpus and affective lexicon should support more comprehensive research in combining cognition grounded data in different NLP tasks.

Other resources for biased emotion model can be extended in two aspects. Firstly, in our proposed a linguistics driven model, the additional morpho-syntactic and orthography features are used in a linear classifier framework using a feature engineering approach.

With the development of neural network model, it is possible to incorporate syntax, semantic and discourse features into neural network models. Secondly, for cognition grounded attention model, our proposed method uses a linear regression model for eye-tracking time prediction. We then build the cognition grounded attention model based on predicted reading time. The two-step pipeline approach may lead to error propagation. And the result of emotion analysis prediction is dependent on the quality of eye-tracking data. In future works, we can explore how to build a unified cognition grounded model to learn the reading time of lexical items and the sentiment label of documents simultaneously. By jointly optimizing reading time prediction and emotion classification , error propagation in the pipeline process can be avoided.

When extending context for node representation learning, we only used text content extended through links. Future work can also explore the use of other type of content including images, and animation videos using a true multi-model approach.

The main purpose of the research detailed in this thesis is to investigate a computational approach to make better use of user profiles for emotion analysis. This thesis studies how to incorporate user profile for emotion analysis systems especially in social media and review text. To sum up, we made progresses in four aspects with significant improvement: (1) improving emotion analysis from cognitions perspective by identifying more appropriate type of linguistic features for our genre of text, (2) using cognition grounded data to improve emotion prediction models, (3) learning the representation of user profiles by addressing the data sparseness through two methods, and (4) incorporating user profiles into emotion analysis model to take subjectivity as a bias into consideration. Incorporating knowledge base and emotion lexicons into emotion analysis model, integrating syntax, semantic and discourse features into neural network models, and building linguistic and learning user profiles though multiple type of content are regarded as three important future directions.

Appendices

Appendix A

Examples of the annotated code-switch dataset

The following table lists some samples of the emotion corpus constructed by us in Chapter 3. The complete corpus can be downloaded.[1]

Table A.1: Samples of built emotion corpus.

Label	**Text**
positive	给对Tingle免疫的人听Tingle？？你好罪哦，怎么酱紫捏！OMG，I love ball pens sounds
positive	表示这里抽到5th Stage的票了，各种激动ing！顺便问下16年1月有没有同行的QAQ?
negative	英文名叫ruby的我心情何等……up主丧[gan]心[de]病[piao]狂[liang].
negative	前100讲道理大家都是纯（hen）洁（wu）的萌（shen）新（shi）求上一次热评呜呜。
positive	Gumi，miku，亚北，天依，rin，大姐，luka，teto，弱音剩下的那个是谁?佐藤莎莎拉?
negative	无端端又被shoot
negative	N站上周不景气，mafu也没有救起来。门口是起分27万的pokemon新番。第一是PPAP动作循环版，75万
positive	我知道了,某幻想通过这个视频间接安利undertale(多好啊我最爱ut了
negative	再说了为啥要care一张膜，反正我是不care这也不代表说什么喜欢滥x交啥的只是一种多余膜的态度罢了!
negative	疯了！买个蛋糕的功夫车竟然打不着了！作啊～CNM,NC

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

Appendix B

Examples of predicted reading time of sentences

B.1 Example in Dundee eye-tracking corpus

The table B.1 is the example of Dundee eye tracking corpus. The whole example sentence is ”The case of Susan Wallace, who went down to her local with Igwig has lesson for us all that go well beyond the blindingly obvious, do not take your igunana to the pub.” In the table B.1, the first six column in the right is the word’s position profile in eye-tracking machine, The last column is the gaze duration (unit:millisecond).

B.2 Example in GECO eye tracking corpus

The following table B.2 lists some samples of the sentence’s eye tracking record in the GECO corpus. The original sentence is ”There was a moment’s stupefied silence. Japp, who was the least surprised of any of us, was the first to speak.” In table B.2, FIX SIZE refers to Fixation Size, fixation locations, fixation durations, temporal order of fixations or scan path, and fixation extent. Fix count refers to fixation count, the times participant gaze at a word.

WORD	TEXT	LINE	OLEN	WLEN	XPOS	WNUM	FDUR
The	1	1	3	3	2	1	80
case	1	1	4	4	8	2	172
of	1	1	2	2	11	3	91
Ms	-99	0	0	0	0	4	0
Susan	1	1	5	5	15	5	207
Wallace,	1	1	8	7	21	6	170
who	-99	0	0	0	0	7	0
went	1	1	4	4	36	8	210
down	1	1	4	4	43	9	184
to	-99	0	0	0	0	10	0
her	-99	0	0	0	0	11	0
local	1	1	5	5	52	12	166
with	1	1	4	4	60	13	196
Igwig	1	1	5	5	66	14	278
the	-99	0	0	0	0	15	0
iguana,	1	1	7	6	75	16	273
has	-99	0	0	0	0	17	0
lessons	-99	0	0	0	0	18	0
for	-99	0	0	0	0	19	0
us	1	2	2	2	17	20	190
all	-99	0	0	0	0	21	0
that	1	2	4	4	23	22	131
go	1	2	2	2	28	23	113
well	1	2	4	4	34	24	182
beyond	1	2	6	6	40	25	174
the	-99	0	0	0	0	26	0
blindingly	1	2	10	10	51	27	153
obvious	1	2	7	7	61	28	148
’do	-99	0	0	0	0	29	0
not	1	2	3	3	72	30	153
take	1	2	4	4	77	31	82
your	-99	0	0	0	0	32	0
iguana	-99	0	0	0	0	33	0
to	1	3	2	2	13	34	128
the	-99	0	0	0	0	35	0
pub’.	1	3	5	3	19	36	218

Table B.1: An example of Dundee eye tracking data

WORD	FIX SIZE	FIX COUNT	1ST RUN STA	1ST RUN END	GAZE DUR
There	2985.5	2	7	359	330
was	.	0	.	.	.
a	2827	1	379	556	177
moment's	2751	1	575	756	181
stupefied	2745	3	786	1651	709
silence.	2812	1	1682	2061	379
Japp,	2719.5	2	2088	2685	445
who	.	0	.	.	.
was	2670	1	2708	2958	250
the	2566	1	3223	3382	159
least	2640	2	2990	3204	214
surprised	2689.5	2	3418	3941	523
of	.	0	.	.	.
any	2723	1	4606	4770	164
of	.	0	.	.	.
us,	2734	1	4854	5058	204
was	.	0	.	.	.
the	2527	1	5213	5411	198
first	2537	1	5436	5620	184
to	2516	1	6007	6092	85
speak.	2501	2	5647	5996	349

Table B.2: An example of GECO corpus

[1]

B.3 Example of predicted reading time of sentences

The table B.3 is the result of predicted eye tracking reading time of selected sentence. The original sentence is "This place is always packed during weekends which tells me this is a great dinner spot."

word	reading time
this	224.3088
place	227.125
is	231.3537
always	240.4325
packed	360.0024
during	234.7833
weekends	272.1682
which	218.7658
tells	291.0104
me	224.1991
this	224.3088
is	231.3537
a	215.3899
great	222.7159
dinner	252.2275
spot	224.9243
.	257.7156

Table B.3: An example of predict reading time in sentences (unit:millisecond)

Appendix C

Examples of predicted values of other affective 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). The affective lexicon value are prediction by using rigid regression model proposed by Li et al.[123]. 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 C.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 C.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

[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`

4	felicific	9.35	5.34	6.83
5	gifts	9.35	6.64	6.71
6	reattend	4.74	4.92	4.68
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

[3]

Table C.2: Examples of extended CVAW 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

[3] (dimensions of Valence-Arousal, Chinese)

35	放松	6.2	2.0
36	闲散	4.6	2.2
37	轻松	6.0	2.2

[4]

Table C.3: Examples of extended EPA 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

[4] (dimensions of Evaluation-Potency-Activity)

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

[5]

Table C.4: Examples of extended DAL 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

[5] (dimensions of Evaluation-Activity-Imagery)

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

[6]

Table C.5: Examples of extended VADER lexicon 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

[6] (dimension of Sentiment)

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

[7]

Table C.6: Examples of extended Perceptual lexicon 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

[7] (dimensions of Hearing-Tasting-Touching-Smelling-Seeing)

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

[8]

Table C.7: Examples of extended Concreteness lexicon 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

[8] (dimension of Concreteness)

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] Ameeta Agrawal and Aijun An. Kea: Sentiment analysis of phrases within short texts. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 380–384. Association for Computational Linguistics, 2014.

[2] Faiyaz Al Zamal, Wendy Liu, and Derek Ruths. Homophily and latent attribute inference: Inferring latent attributes of twitter users from neighbors. *ICWSM*, 270, 2012.

[3] 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. Association for Computational Linguistics.

[4] Hadi Amiri and Tat-Seng Chua. Mining slang and urban opinion words and phrases from cqa services: an optimization approach. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 193–202. ACM, 2012.

[5] William N Anderson Jr and Thomas D Morley. Eigenvalues of the laplacian of a graph. *Linear and multilinear algebra*, 18(2):141–145, 1985.

[6] Ask Me Anything. Dynamic memory networks for natural language processing. *Kumar et al. arXiv Pre-Print*, 2015.

[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] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. pages 3104–3112, 2014.

[9] AR Balamurali, Aditya Joshi, and Pushpak Bhattacharyya. Harnessing wordnet senses for supervised sentiment classification. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1081–1091. Association for Computational Linguistics, 2011.

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

[11] Themis Balomenos, Amaryllis Raouzaiou, Spiros Ioannou, Athanasios Drosopoulos, Kostas Karpouzis, and Stefanos Kollias. Emotion analysis in man-machine interaction systems. In *International Workshop on Machine Learning for Multimodal Interaction*, pages 318–328. Springer, 2004.

[12] Luciano Barbosa and Junlan Feng. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 36–44. Association for Computational Linguistics, 2010.

[13] Maria Barrett, Joachim Bingel, Frank Keller, and Anders Søgaard. Weakly supervised part-of-speech tagging using eye-tracking data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics: Short Papers*, volume 579, page 584. Association for Computational Linguistics, 2016.

[14] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*, pages 585–591. International Machine Learning Society (IMLS), 2002.

[15] Farah Benamara, Carmine Cesarano, Antonio Picariello, Diego Reforgiato, and V. S. Subrahmanian. Sentiment analysis: Adjectives and adverbs are better than adjectives alone. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*, 2007. Short paper.

[16] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *journal of machine learning research*, 3(Feb):1137–1155, 2003.

[17] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[18] Tanja Blascheck, Kuno Kurzhals, Michael Raschke, Michael Burch, Daniel Weiskopf, and Thomas Ertl. State-of-the-art of visualization for eye tracking data. In *Proceedings of EuroVis 2014*, volume 2014, 2014.

[19] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[20] Eric Bloedorn and Inderjeet Mani. Using nlp for machine learning of user profiles. *Intelligent Data Analysis*, 2(1):3–18, 1998.

[21] 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.

[22] 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.

[23] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[24] 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.

[25] John D Burger, John Henderson, George Kim, and Guido Zarrella. Discriminating gender on twitter. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1301–1309. Association for Computational Linguistics, 2011.

[26] 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.

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

[28] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Deep neural networks for learning graph representations. In *AAAI*, pages 1145–1152, 2016.

[29] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27, 2011.

[30] Shiyu Chang, Wei Han, Jiliang Tang, Guo-Jun Qi, Charu C Aggarwal, and Thomas S Huang. Heterogeneous network embedding via deep architectures. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 119–128. ACM, 2015.

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

[32] Wanxiang Che, Valentin I Spitkovsky, and Ting Liu. A comparison of chinese parsers for stanford dependencies. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 11–16. Association for Computational Linguistics, 2012.

[33] Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. Neural sentiment classification with user and product attention. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 1660–1670. Association for Computational Linguistics, 2016.

[34] I-Hsuan Chen, Yunfei Long, Qin Lu, and Chu-Ren Huang. Leveraging eventive information for better metaphor detection and classification. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 36–46, 2017.

[35] Minmin Chen. Efficient vector representation for documents through corruption. *arXiv preprint arXiv:1707.02377*, 2017.

[36] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[37] Morgane Ciot, Morgan Sonderegger, and Derek Ruths. Gender inference of twitter users in non-english contexts. In *EMNLP*, pages 1136–1145, 2013.

[38] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649. IEEE, 2012.

[39] Michael D Conover, Bruno Gonçalves, Jacob Ratkiewicz, Alessandro Flammini, and Filippo Menczer. Predicting the political alignment of twitter users. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, pages 192–199. IEEE, 2011.

[40] Uschi Cop, Nicolas Dirix, Denis Drieghe, and Wouter Duyck. Presenting geco: An eyetracking corpus of monolingual and bilingual sentence reading. *Behavior research methods*, pages 1–14, 2016.

[41] Ayse Cufoglu. User profiling-a short review. *International Journal of Computer Applications*, 108(3), 2014.

[42] Ido Dagan, Shaul Marcus, and Shaul Markovitch. Contextual word similarity and estimation from sparse data. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pages 164–171. Association for Computational Linguistics, 1993.

[43] George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):30–42, 2012.

[44] Rajarshi Das, Manzil Zaheer, Siva Reddy, and Andrew McCallum. Question answering on knowledge bases and text using universal schema and memory networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 358–365, 2017.

[45] Dmitry Davidov, Oren Tsur, and Ari Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of the 23rd international conference on computational linguistics: posters*, pages 241–249. Association for Computational Linguistics, 2010.

[46] Steven Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*, 28(4):357–366, 1980.

[47] Vera Demberg and Frank Keller. Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition*, 109(2):193–210, 2008.

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

[49] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 193–202. ACM, 2014.

[50] Chuong B Do and Andrew Y Ng. Transfer learning for text classification. In *Advances in Neural Information Processing Systems*, pages 299–306, 2006.

[51] Cicero dos Santos and Maira Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78, 2014.

[52] Zi-Yi Dou. Capturing user and product information for document level sentiment analysis with deep memory network. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 521–526. Association for Computational Linguistics, 2017.

[53] Susan T Dumais. Latent semantic analysis. *Annual review of information science and technology*, 38(1):188–230, 2004.

[54] Almudena Duque and Carmelo Vázquez. Double attention bias for positive and negative emotional faces in clinical depression: Evidence from an eye-tracking study. *Journal of Behavior Therapy and Experimental Psychiatry*, 46:107–114, 2015.

[55] Paul Ekman. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200, 1992.

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

[57] Paul Ekman, Robert W Levenson, and Wallace V Friesen. Autonomic nervous system activity distinguishes among emotions. *Science*, 221(4616):1208–1210, 1983.

[58] John L Fischer. Social influences on the choice of a linguistic variant. *Word*, 14(1):47–56, 1958.

[59] CJ Hutto Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International Conference on Weblogs and Social Media (ICWSM-14). Available at (20/04/16) http://comp. social. gatech. edu/papers/icwsm14. vader. hutto. pdf*, 2014.

[60] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1(2009):12, 2009.

[61] Christoph Goller and Andreas Kuchler. Learning task-dependent distributed representations by backpropagation through structure. In *Neural Networks, 1996., IEEE International Conference on*, volume 1, pages 347–352. IEEE, 1996.

[62] Gene Golub and William Kahan. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, 2(2):205–224, 1965.

[63] Wei Gong, Ee-Peng Lim, and Feida Zhu. Characterizing silent users in social media communities. In *ICWSM*, pages 140–149, 2015.

[64] Swapna Gottipati, Minghui Qiu, Liu Yang, Feida Zhu, and Jing Jiang. Predicting user's political party using ideological stances. In *Proceedings of the 5th International Conference on Social Informatics*, pages 177–191. Springer, 2013.

[65] Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *arXiv preprint arXiv:1705.02801*, 2017.

[66] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. Association for Computing Machinery, 2016.

[67] Lin Gui, Ruifeng Xu, Yulan He, Qin Lu, and Zhongyu Wei. Intersubjectivity and sentiment: from language to knowledge. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 2789–2795. Association for the Advancement of Artificial Intelligence, 2016.

[68] Weiwei Guo and Mona Diab. Modeling sentences in the latent space. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 864–872. Association for Computational Linguistics, 2012.

[69] Weiwei Guo and Mona T Diab. Improving lexical semantics for sentential semantics: Modeling selectional preference and similar words in a latent variable model. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 739–745. Association for Computational Linguistics, 2013.

[70] Yoan Gutiérrez, Sonia Vázquez, and Andrés Montoyo. Sentiment classification using semantic features extracted from wordnet-based resources. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, pages 139–145. Association for Computational Linguistics, 2011.

[71] Michael Hahn and Frank Keller. Modeling human reading with neural attention. pages 85–95.

[72] Hussam Hamdan, Frederic Béchet, and Patrice Bellot. Experiments with dbpedia, wordnet and sentiwordnet as resources for sentiment analysis in micro-blogging. 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 455–459. Association for Computing Machinery, 2013.

[73] 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.

[74] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

[75] Vasileios Hatzivassiloglou and Kathleen R McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 174–181. Association for Computational Linguistics, 1997.

[76] Vasileios Hatzivassiloglou and Janyce M Wiebe. Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of the 18th conference*

on Computational linguistics-Volume 1, pages 299–305. Association for Computational Linguistics, 2000.

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

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

[79] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

[80] Wenxing Hong, Siting Zheng, and Huan Wang. Dynamic user profile-based job recommender system. In *Computer Science & Education (ICCSE), 2013 8th International Conference on*, pages 1499–1503. IEEE, 2013.

[81] Yu-Lun Hsieh, Shih-Hung Liu, Yung-Chun Chang, and Wen-Lian Hsu. Neural network-based vector representation of documents for reader-emotion categorization. In *Information Reuse and Integration (IRI), 2015 IEEE International Conference on*, pages 569–573. IEEE, 2015.

[82] Liang Hu, Guohang Song, Zhenzhen Xie, and Kuo Zhao. Personalized recommendation algorithm based on preference features. *Tsinghua Science and Technology*, 19(3):293–299, 2014.

[83] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272. IEEE, 2008.

[84] Chu-Ren Huang and Dingxu Shi. *A reference grammar of Chinese*. Cambridge University Press, 2016.

[85] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154, 1962.

[86] Eva Hudlicka. What are we modeling when we model emotion? In *AAAI spring symposium: emotion, personality, and social behavior*, volume 8, pages 190–197, 2008.

[87] Carroll W Hughes. Emotion: Theory, research and experience. *The Journal of Nervous and Mental Disease*, 170(5):315–316, 1982.

[88] Adrian Iftene and Jean Vanderdonckt. Moocbuddy: a chatbot for personalized learning with moocs. In *RoCHI–International Conference on Human-Computer Interaction*, page 91, 2016.

[89] Ozan Irsoy and Claire Cardie. Opinion mining with deep recurrent neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 720–728. Association for Computational Linguistics, 2014.

[90] Derek M Isaacowitz, Heather A Wadlinger, Deborah Goren, and Hugh R Wilson. Selective preference in visual fixation away from negative images in old age? an eye-tracking study. *Psychology and aging*, 21(1):40, 2006.

[91] Katherine Isbister and Florian "Floyd" Mueller. Guidelines for the design of movement-based games and their relevance to hci. *Human–Computer Interaction*, 30(3-4):366–399, 2015.

[92] Carroll E Izard. Emotion theory and research: Highlights, unanswered questions, and emerging issues. *Annual review of psychology*, 60:1–25, 2009.

[93] J Edward Jackson. *A user's guide to principal components*, volume 587. John Wiley & Sons, 2005.

[94] Rubén Jacob-Dazarola, Juan Carlos Ortíz Nicolás, and Lina Cárdenas Bayona. 5 - behavioral measures of emotion. In Herbert L. Meiselman, editor, *Emotion Measurement*, pages 101–124. Woodhead Publishing, 2016.

[95] Sarthak Jain. Question answering over knowledge base using factual memory networks. In *Proceedings of the NAACL Student Research Workshop*, pages 109–115, 2016.

[96] Bernard J Jansen, Mimi Zhang, Kate Sobel, and Abdur Chowdury. Twitter power: Tweets as electronic word of mouth. *Journal of the American society for information science and technology*, 60(11):2169–2188, 2009.

[97] Philip Nicholas Johnson-Laird and Keith Oatley. The language of emotions: An analysis of a semantic field. *Cognition and emotion*, 3(2):81–123, 1989.

[98] Aditya Joshi, Abhijit Mishra, Nivvedan Senthamilselvan, and Pushpak Bhattacharyya. Measuring sentiment annotation complexity of text. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 36–41, 2014.

[99] Neha S Joshi and Suhasini A Itkat. A survey on feature level sentiment analysis. *International Journal of Computer Science and Information Technologies*, 5:5422–5425, 2014.

[100] Michael Hahn Frank Keller. Modeling human reading with neural attention. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, volume 85, page 95. Association of Computational Linguistics, 2016.

[101] A Kennedy. The dundee corpus [cd-rom]. *Psychology Department, University of Dundee*, 2003.

[102] Adam Kilgarriff. Wordnet: An electronic lexical database, 2000.

[103] Geunyoung Kim, Tedra Walden, Vicki Harris, Jan Karrass, and Thomas Catron. Positive emotion, negative emotion, and emotion control in the externalizing problems of school-aged children. *Child psychiatry and human development*, 37(3):221–239, 2007.

[104] Soo-Min Kim and Eduard Hovy. Determining the sentiment of opinions. In *Proceedings of the 20th international conference on Computational Linguistics*, page 1367. Association for Computational Linguistics, 2004.

[105] Yoon Kim. Convolutional neural networks for sentence classification. 2014.

[106] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[107] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302, 2015.

[108] Seema Kolkur, Gayatri Dantal, and Reena Mahe. Study of different levels for sentiment analysis. *International Journal of Current Engineering and Technology*, 5(2):768Ű770, 2015.

[109] Svetlana Kordumova, Jan van Gemert, and Cees GM Snoek. Exploring the long tail of social media tags. In *International Conference on Multimedia Modeling*, pages 51–62. Springer, 2016.

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

[111] William Labov. *The social stratification of English in New York city*. Cambridge University Press, 2006.

[112] Angeliki Lazaridou, Marco Marelli, Roberto Zamparelli, and Marco Baroni. Compositional-ly derived representations of morphologically complex words in distributional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1517–1526, 2013.

[113] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*, 2014.

[114] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, volume 14, pages 1188–1196, 2014.

[115] Sophia Lee and Zhongqing Wang. Emotion in code-switching texts: Corpus construction and analysis. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing*, pages 91–99. Association for Computational Linguistics, 2015.

[116] Kristina Lerman and Rumi Ghosh. Information contagion: An empirical study of the spread of news on digg and twitter social networks. *International AAAI Conference on Web and Social Media (ICWSM)*, 10:90–97, 2010.

[117] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 177–187. ACM, 2005.

[118] 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. Association for Computational Linguistics, 2014.

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

[120] Changliang Li, Bo Xu, Gaowei Wu, Saike He, Guanhua Tian, and Hongwei Hao. Recursive deep learning for sentiment analysis over social data. In *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 02*, pages 180–185. IEEE Computer Society, 2014.

[121] Minglei Li. *Emotion analysis from text*. PhD thesis, The Hong Kong Polytechnic University, 2018.

[122] 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.

[123] Minglei Li, Qin Lu, Yunfei Long, and Lin Gui. Inferring affective meanings of words from word embedding. *IEEE Transactions on Affective Computing*, 2017.

[124] Shoushan Li, Sophia Yat Mei Lee, Ying Chen, Chu-Ren Huang, and Guodong Zhou. Sentiment classification and polarity shifting. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 635–643. Association for Computational Linguistics, 2010.

[125] Chenghua Lin and Yulan He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 375–384. ACM, 2009.

[126] Rui Lin, Shujie Liu, Muyun Yang, Mu Li, Ming Zhou, and Sheng Li. Hierarchical recurrent neural network for document modeling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 899–907. Association for Computational Linguistics, 2015.

[127] Sea Ling, Maria Indrawan, and Seng W Loke. Rfid-based user profiling of fashion preferences: blueprint for a smart wardrobe. *International Journal of Internet Protocol Technology*, 2(3-4):153–164, 2007.

[128] Nedim Lipka and Benno Stein. Classifying with co-stems. In *European Conference on Information Retrieval*, pages 307–313. Springer, 2011.

[129] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*. John Wiley & Sons, 2014.

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

[131] Bing Liu and Lei Zhang. A survey of opinion mining and sentiment analysis. In *Mining text data*, pages 415–463. Springer, 2012.

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

[133] Yunfei Long, Qin Lu, Yue Xiao, MingLei Li, and Chu-Ren Huang. Domain-specific user preference prediction based on multiple user activities. In *Big Data (Big Data), 2016 IEEE International Conference on*, pages 3913–3921. Institute of Electrical and Electronics Engineers, 2016.

[134] Yunfei Long, Lu Qin, Rong Xiang, Minglei Li, and Chu-Ren Huang. A cognition based attention model for sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 473–482, 2017.

[135] Julie Beth Lovins. Development of a stemming algorithm. *Mech. Translat. & Comp. Linguistics*, 11(1-2):22–31, 1968.

[136] Dijun Luo, Feiping Nie, Heng Huang, and Chris H Ding. Cauchy graph embedding. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 553–560. Association for Computing Machinery, 2011.

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

[138] 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.

[139] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 931–940. Association for Computing Machinery, 2008.

[140] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[141] Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 142–150. Association for Computational Linguistics, 2011.

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

[143] Ronald KS Macaulay. *Talk that counts: Age, gender, and social class differences in discourse*. Oxford University Press, 2005.

[144] Suraj Maharjan, Elizabeth Blair, Steven Bethard, and Thamar Solorio. Developing language-tagged corpora for code-switching tweets. In *Proceedings of The 9th Linguistic Annotation Workshop*, pages 72–84. Association for Computational Linguistics, 2015.

[145] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (System Demonstrations)*, pages 55–60, 2014.

[146] Justin Martineau and Tim Finin. Delta tfidf: An improved feature space for sentiment analysis. *Icwsm*, 9:106, 2009.

[147] Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.

[148] Scott McDonald. The long tail and its implications for media audience measurement. *Journal of Advertising Research*, 48(3):313–319, 2008.

[149] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.

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

[151] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web*, pages 171–180. Association for Computing Machinery, 2007.

[152] Igor Mel'cuk and Leo Wanner. Lexical functions and lexical inheritance for emotion lexemes in german. *Lexical functions in lexicography and natural language processing*, 31:209, 1996.

[153] Ralf Metzler and Joseph Klafter. The restaurant at the end of the random walk: recent developments in the description of anomalous transport by fractional dynamics. *Journal of Physics A: Mathematical and General*, 37(31):R161, 2004.

[154] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *Proceedings of Workshop at International Conference on Learning Representations*, 2013.

[155] Abhijit Mishra and Pushpak Bhattacharyya. Predicting readers' sarcasm understandability by modeling gaze behavior. In *Cognitively Inspired Natural Language Processing*, pages 99–115. Springer, 2018.

[156] Abhijit Mishra, Pushpak Bhattacharyya, Michael Carl, and IBC CRITT. Automatically predicting sentence translation difficulty. In *ACL (2)*, pages 346–351, 2013.

[157] Abhijit Mishra, Kuntal Dey, and Pushpak Bhattacharyya. Learning cognitive features from gaze data for sentiment and sarcasm classification using convolutional neural network. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 377–387, 2017.

[158] Abhijit Mishra, Diptesh Kanojia, and Pushpak Bhattacharyya. Predicting readers' sarcasm understandability by modeling gaze behavior. In *AAAI*, pages 3747–3753, 2016.

[159] Abhijit Mishra, Diptesh Kanojia, Seema Nagar, Kuntal Dey, and Pushpak Bhattacharyya. Leveraging cognitive features for sentiment analysis. *CoNLL 2016*, page 156, 2016.

[160] Abhijit Mishra, Srikanth Tamilselvam, Riddhiman Dasgupta, Seema Nagar, and Kuntal Dey. Cognition-cognizant sentiment analysis with multitask subjectivity summarization based on annotators' gaze behavior. 2018.

[161] 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.

[162] Saif M Mohammad. Sentiment analysis: Detecting valence, emotions, and other affectual states from text. In *Emotion measurement*, pages 201–237. Elsevier, 2016.

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

[164] 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.

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

[166] Tony Mullen and Nigel Collier. Sentiment analysis using support vector machines with diverse information sources. In *EMNLP*, volume 4, pages 412–418, 2004.

[167] Tetsuji Nakagawa, Kentaro Inui, and Sadao Kurohashi. Dependency tree-based sentiment classification using crfs with hidden variables. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 786–794. Association for Computational Linguistics, 2010.

[168] Richard E Nisbett and Timothy D Wilson. The halo effect: Evidence for unconscious alteration of judgments. *Journal of personality and social psychology*, 35(4):250, 1977.

[169] Andrew Ortony, Gerald L Clore, and Allan Collins. *The cognitive structure of emotions*. Cambridge university press, 1990.

[170] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1105–1114. Association for Computing Machinery, 2016.

[171] Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. Unsupervised learning of sentence embeddings using compositional n-gram features. *arXiv preprint arXiv:1703.02507*, 2017.

[172] Shirui Pan, Jia Wu, Xingquan Zhu, Chengqi Zhang, and Yang Wang. Tri-party deep network representation. 11(9):12, 2016.

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

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

[175] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.

[176] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[177] John M Pearce. A model for stimulus generalization in pavlovian conditioning. *Psychological review*, 94(1):61, 1987.

[178] Marco Pennacchiotti and Ana-Maria Popescu. A machine learning approach to twitter user classification. *Icwsm*, 11(1):281–288, 2011.

[179] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceeds of 2018 Conference on Empirical Methods in Natural Language Processing*, volume 14, pages 1532–1543. Association for Computational Linguistics, 2014.

[180] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. Association for Computing Machinery, 2014.

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

[182] Rosalind W Picard. Affective computing: challenges. *International Journal of Human-Computer Studies*, 59(1-2):55–64, 2003.

[183] Qiao Qian, Minlie Huang, Jinhao Lei, and Xiaoyan Zhu. Linguistically regularized lstms for sentiment classification. *arXiv preprint arXiv:1611.03949*, 2016.

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

[185] Delip Rao and David Yarowsky. Detecting latent user properties in social media. In *Proc. of the NIPS MLSN Workshop*. Citeseer, 2010.

[186] 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.

[187] Keith Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 124(3):372, 1998.

[188] Ellen Riloff and Janyce Wiebe. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 105–112. Association for Computational Linguistics, 2003.

[189] Ira J Roseman. A model of appraisal in the emotion system: Integrating theory, research, and applications. pages 68–91, 2001.

[190] Sara Rosenthal, Noura Farra, and Preslav Nakov. Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518. Institute of Electrical and Electronics Engineers, 2017.

[191] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

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

[193] Mayur Rustagi, R Rajendra Prasath, Sumit Goswami, and Sudeshna Sarkar. Learning age and gender of blogger from stylistic variation. In *International Conference on Pattern Recognition and Machine Intelligence*, pages 205–212. Springer, 2009.

[194] 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. Association for Computational Linguistics, The Association for Computer Linguistics.

[195] Cicero D Santos and Bianca Zadrozny. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1818–1826, 2014.

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

[197] Klaus R Scherer and Harald G Wallbott. Evidence for universality and cultural variation of differential emotion response patterning. *Journal of personality and social psychology*, 66(2):310, 1994.

[198] 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.

[199] Kim Schouten and Flavius Frasincar. Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):813–830, 2016.

[200] Blake Shaw and Tony Jebara. Structure preserving embedding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 937–944. Association for Computing Machinery, 2009.

[201] 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 (ICML-11)*, pages 129–136, 2011.

[202] Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the conference on empirical methods in natural language processing*, pages 151–161. Association for Computational Linguistics, 2011.

[203] 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 conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642. Citeseer, 2013.

[204] 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.

[205] Harald Steck. Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 713–722. Association for Computing Machinery, 2010.

[206] Jan E Stets. Emotions and sentiments. In *Handbook of social psychology*, pages 309–335. Springer, 2006.

[207] Philip J Stone, Dexter C Dunphy, and Marshall S Smith. The general inquirer: A computer approach to content analysis. 1966.

[208] Carlo Strapparava and Rada Mihalcea. Semeval-2007 task 14: Affective text. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 70–74. Association for Computational Linguistics, 2007.

[209] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448, 2015.

[210] Xiaofei Sun, Jiang Guo, Xiao Ding, and Ting Liu. A general framework for content-enhanced network representation learning. *arXiv preprint arXiv:1610.02906*, 2016.

[211] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[212] Feride Savaroğlu Tabak and Vesile Evrim. Comparison of emotion lexicons. In *HONET-ICT, 2016*, pages 154–158. IEEE, 2016.

[213] Chenhao Tan, Lillian Lee, Jie Tang, Long Jiang, Ming Zhou, and Ping Li. User-level sentiment analysis incorporating social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1397–1405. ACM, 2011.

[214] Duyu Tang, Bing Qin, and Ting Liu. Learning semantic representations of users and products for document level sentiment classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1014–1023. Association for Computational Linguistics, 2015.

[215] Duyu Tang, Bing Qin, and Ting Liu. Learning semantic representations of users and products for document level sentiment classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1014–1023, Beijing, China, July 2015. Association for Computational Linguistics.

[216] Duyu Tang, Bing Qin, and Ting Liu. Aspect level sentiment classification with deep memory network. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 214–225, 2016.

[217] 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.

[218] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1555–1565. Association for Computational Linguistics, 2014.

[219] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. Association for Computing Machinery, 2015.

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

[221] Loren Terveen and Will Hill. Beyond recommender systems: Helping people help each other. *HCI in the New Millennium*, 1:487–509, 2001.

[222] Katrin Tomanek, Udo Hahn, Steffen Lohmann, and Jürgen Ziegler. A cognitive cost model of annotations based on eye-tracking data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1158–1167. Association for Computational Linguistics, 2010.

[223] Richard Tong. An operational system for detecting and tracking opinions in online discussions. In *Working Notes of the SIGIR Workshop on Operational Text Classification*, pages 1–6, New Orleans, Louisianna, 2001.

[224] Cunchao Tu, Han Liu, Zhiyuan Liu, and Maosong Sun. Cane: Context-aware network embedding for relation modeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1722–1731. Association for Computational Linguistics, 2017.

[225] Cunchao Tu, Hao Wang, Xiangkai Zeng, Zhiyuan Liu, and Maosong Sun. Community-enhanced network representation learning for network analysis. *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 2011–2018, 2016.

[226] Cunchao Tu, Weicheng Zhang, Zhiyuan Liu, and Maosong Sun. Max-margin deepwalk: Discriminative learning of network representation. In *IJCAI*, pages 3889–3895, 2016.

[227] Peter D Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics, 2002.

[228] 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.

[229] 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.

[230] David Vilares, Miguel A Alonso, and Carlos Gómez-Rodríguez. Sentiment analysis on monolingual, multilingual and code-switching twitter corpora. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 2–8. Association for Computational Linguistics, 2015.

[231] David Vilares, Miguel A Alonso, and Carlos Gómez-Rodríguez. En-es-cs: An english-spanish code-switching twitter corpus for multilingual sentiment analysis. In *LREC*, 2016.

[232] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1225–1234. Association for Computing Machinery, 2016.

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

[234] Shih-Ming Wang and Lun-Wei Ku. Antusd: A large chinese sentiment dictionary. In *Proceedings of the 2016 International Conference on Language Resources and Evaluation*. ELRA, 2016.

[235] William Yang Wang. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 422–426, 2017.

[236] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of 2016 Conference on Empirical Methods in Natural Language Processing*, pages 606–615, 2016.

[237] Zhongqing Wang, Sophia Yat Mei Lee, Shoushan Li, and Guodong Zhou. Emotion analysis in code-switching text with joint factor graph model. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(3):469–480, 2017.

[238] 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.

[239] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. In *Proceedings of the 2016 Association for Computational Linguistics*, 2014.

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

[241] Janyce Wiebe. Learning subjective adjectives from corpora. *Proceedings of the Seventeens National Conference on Artificial Intelligence*, 20(0):735–740, 2000.

[242] Janyce Wiebe, Rebecca Bruce, Matthew Bell, Melanie Martin, and Theresa Wilson. A corpus study of evaluative and speculative language. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue-Volume 16*, pages 1–10. Association for Computational Linguistics, 2001.

[243] Theresa Wilson, Zornitsa Kozareva, Preslav Nakov, Sara Rosenthal, Veselin Stoyanov, and Alan Ritter. Sentiment analysis in twitter. In *Proceedings of the International Workshop on Semantic*, 2013.

[244] 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.

[245] Jiacheng Xu, Danlu Chen, Xipeng Qiu, and Xuangjing Huang. Cached long short-term memory neural networks for document-level sentiment classification. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.

[246] Linhong Xu, Hongfei Lin, Yu Pan, Hui Ren, and Jianmei Chen. Constructing the affective lexicon ontology. *Journal of the China Society for Scientific and Technical Information*, 27(2):180–185, 2008.

[247] Ruifeng Xu, Chengtian Zou, Jun Xu, and Q Lu. Reader's emotion prediction based on partitioned latent dirichlet allocation model. In *Proceedings of of International Conference on Internet Computing and Big Data*. Association for Computing Machinery, 2013.

[248] Jasy Liew Suet Yan, Howard R Turtle, and Elizabeth D Liddy. Emotweet-28: a fine-grained emotion corpus for sentiment analysis. In *Proceedings of the 10th International Conference on Language Resources and Evaluation. LREC*, pages 1149–1156, 2016.

[249] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. Network representation learning with rich text information. In *IJCAI*, pages 2111–2117, 2015.

[250] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. Network representation learning with rich text information. In *IJCAI*, pages 2111–2117, 2015.

[251] 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.

[252] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016.

[253] 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.

[254] 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. Association for Computational Linguistics, 2015.

[255] Lei Zhang, Shuai Wang, and Bing Liu. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, page e1253, 2018.

[256] 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.