Linguistic-inspired Chinese sentiment analysis : from characters to radicals and phonetics

Peng, Haiyun

2019

Peng, H. (2019). Linguistic-inspired Chinese sentiment analysis : from characters to radicals and phonetics. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/84297>

<https://doi.org/10.32657/10220/48173>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

**LINGUISTIC-INSPIRED
CHINESE SENTIMENT ANALYSIS:
FROM CHARACTERS
TO RADICALS AND PHONETICS**

HAIYUN PENG

**SCHOOL OF COMPUTER SCIENCE AND
ENGINEERING**

A thesis submitted to the Nanyang Technological University in partial
fulfillment of the requirement for the degree of Doctor of Philosophy

2019

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

May 2, 2019

Haiyun Peng

A handwritten signature in black ink, reading "Peng Haiyun". The signature is written in a cursive, flowing style. The first name "Peng" is written with a large, stylized 'P' and the last name "Haiyun" follows in a similar cursive script.

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

May 2, 2019

Erik Cambria

A handwritten signature in blue ink, appearing to read "Erik Cambria", with a stylized flourish at the end.

Authorship Attribution Statement

This thesis contains material from 4 papers published in the following peer-reviewed journals / from papers accepted at conferences in which I am listed as an author.

Chapter 2 is published as Haiyun Peng, Erik Cambria, and Amir Hussain. "A review of sentiment analysis research in Chinese language." *Cognitive Computation* 9, no. 4 (2017): 423-435.

The contributions of the co-authors are as follows:

- Prof Cambria suggested the review area and edited the manuscript drafts.
- I reviewed the literature and wrote the review manuscript draft.
- Prof Hussain proofread the manuscript.

Chapter 3 is published as Haiyun Peng and Erik Cambria. "CSenticNet: A Concept-level Resource for Sentiment Analysis in Chinese Language." In *International Conference on Computational Linguistics and Intelligent Text Processing (CiCLing)*, 90-104, 2017.

The contributions of the co-authors are as follows:

- Prof Cambria suggested topic, edited and proofread the paper.
- I designed the algorithm, conducted experiments and wrote the paper.

Chapter 4 is published as Haiyun Peng, Erik Cambria, and Xiaomei Zou. "Radical-based hierarchical embeddings for chinese sentiment analysis at sentence level." In *FLAIRS conference*, 347-352, 2017.

The contributions of the co-authors are as follows:

- Prof Cambria participated in discussion, edited and proofread the paper.
- I designed the methodology and conducted the experiments and wrote the paper.
- Xiaomei processed parsing Chinese characters to Chinese radicals.

Chapter 5 is published as Haiyun Peng, Yukun Ma, Yang Li, and Erik Cambria. Learning multigrained aspect target sequence for Chinese sentiment analysis.” Knowledge-Based Systems 148 (2018): 167-176.

The contributions of the co-authors are as follows:

- Prof Cambria participated in discussion, edited and proofread the manuscript.
- I designed the models, ran the experiments, and wrote the manuscript.
- Yukun participated in discussion and helped design experimental validation.
- Yang participated in discussion and extracted visual features.

May 2, 2019

Haiyun Peng

A handwritten signature in black ink, reading "Peng Haiyun". The signature is written in a cursive, flowing style. The first character "Peng" is larger and more prominent, followed by "Haiyun" in a similar but slightly smaller script. The ink is dark and the background is white.

Acknowledgments

First of all, I would like to express my sincere gratitude towards my PhD supervisor Prof. Erik Cambria. For the past four years, he has been continuously supportive and encouraging. Without his patient and insightful guidance, I would not have acquired the knowledge and skills to reach this stage.

I would like to thank my TAC panel members and Co-supervisor, Prof. Quek Hiok Chai, Prof. Francis Bond and Dr. Chi Xu for their helpful comments and advice.

In addition, I would like to thank Dr. Soujanya Poria and Dr. Yukun Ma for their supportive and inspiring discussions and assistance during my PhD study. My study would not have been complete without their collaborations.

Furthermore, my PhD journey would not end up as a pleasant and rewarding journey without the friendship and help by Sa Gao, Dr. Yang Li, Sandro Cavallari, Qian Chen, Edoardo Ragusa, Pranav Rai and Xiaomei Zou.

I would like to thank my girlfriend, Xiuhua Geng, for all her love and support.

Last but not the least, I would like to express my deep gratitude towards my parents for all the love and support. I am especially thankful to my mother, Ying Chen, for her understanding and relentless support during the crucial stages of my life. This thesis is completely dedicated to them.

Contents

Statement of Originality	i
Supervisor Declaration Statement	ii
Authorship Attribution Statement	iii
Acknowledgments	v
List of Figures	ix
List of Tables	xi
Abstract	
1 Introduction	1
1.1 Background	1
1.2 Challenges	5
1.3 Contributions	7
1.4 Organization	8
2 Literature Review	9
2.1 Sentiment Resource	9
2.1.1 Corpus	9
2.1.2 Lexicon	11
2.2 Monolingual Approach	13
2.2.1 Preprocessing	13
2.2.2 Machine Learning-based Approach	14
2.2.3 Knowledge-based Approach	18
2.2.4 Mix Models	20
2.3 Multilingual Approach	21
2.4 Text Representation	22
2.4.1 General Embedding Methods	22
2.4.2 Chinese Text Representation	23
2.5 Chinese Phonology	24
2.6 Summary	25

3	CsenticNet	26
3.1	Introduction	26
3.2	Background	27
3.3	Overview	28
3.3.1	Resources	28
3.3.2	Two Versions	29
3.4	First Version: SentiWordNet + NTU MC	30
3.5	Second Version: SenticNet + NTU MC	32
3.5.1	SenticNet and Preprocessing	32
3.5.2	Mapping SenticNet to WordNet	32
3.5.3	Find and Extract the Overlap	36
3.6	Evaluation	36
3.7	Summary	39
4	Radical-Based Hierarchical Embeddings	40
4.1	Introduction	40
4.2	Background	41
4.2.1	Chinese Radicals	42
4.3	Hierarchical Chinese Embedding	43
4.3.1	Skip-Gram Model	44
4.3.2	Radical-Based Embedding	45
4.3.3	Hierarchical Embedding	46
4.4	Experimental Evaluation	47
4.4.1	Dataset	47
4.4.2	Experimental Setting	48
4.4.3	Results and Discussion	48
4.5	Summary	50
5	Multi-grained Aspect Target Sequence Modeling	51
5.1	Introduction	51
5.2	Background	54
5.3	Method Overview	55
5.3.1	Aspect Target Sequence	55
5.3.2	Task Definition	55
5.3.3	Overview of the Algorithm	56
5.4	Adaptive Embedding Learning	57

5.4.1	Sentence Sequence Learning	57
5.4.2	Aspect Target Unit Learning	58
5.5	Sequence Learning of Aspect Target	59
5.6	Fusion of Multi-Granularity Representation	59
5.6.1	Early Fusion	61
5.6.2	Late Fusion	61
5.7	Evaluation	63
5.7.1	Datasets	63
5.7.2	Comparison Methods	64
5.7.3	Result Analysis	67
5.7.4	Visual Case Study	69
5.7.5	Granularity and Fusion Analysis	70
5.8	Summary	73
6	Phonetic-enriched Text Representation	74
6.1	Introduction	74
6.2	Model	76
6.2.1	Textual Embedding	76
6.2.2	Training Visual Features	77
6.2.3	Learning Phonetic Features	78
6.2.4	DISA	83
6.2.5	Fusion of Modalities	87
6.3	Experiments and Results	88
6.3.1	Experimental Setup	88
6.3.2	Experiments on Unimodality	91
6.3.3	Experiments on Fusion of Modalities	93
6.3.4	Cross-domain Evaluation	94
6.3.5	Ablation Tests	96
6.3.6	Visualization	99
6.4	Summary	101
7	Summary and Future Work	102
7.1	Summary of Proposed Method	102
7.2	Limitations and Future Work	104

List of Figures

1.1	Syntax trees for the sentence “Everything would have been all right if you hadn’t said that” in two languages	2
1.2	Example of importance of word segmentation	3
1.3	Research path and motivation	7
2.1	Hourglass model in the Chinese language	13
2.2	Machine learning-based processing for Chinese sentiment analysis . . .	14
2.3	Evolution of NLP research through three different eras from [1]	20
3.1	CSenticNet	28
3.2	Example of SenticNet semantic graph.	29
3.3	Example of used sentiment resources	31
3.4	Mapping Framework of SenticNet Version	36
3.5	Distribution of sentiment values	38
4.1	Performance on four datasets at different fusion parameter	46
4.2	Framework of hierarchical embedding model	47
5.1	ATSM-F late fusion framework. RNN-1,-2,-3 are at word, character and radical level, respectively. Green RNNs are for adaptive embedding learning. Grey RNNs are sequence learning of aspect target. Aspect target is highlighted in red.	56
5.2	Fusion mechanisms.	62
5.3	Visual attention weights of each word in the example. (a) is from ATSM-S. (b) is from baseline model.	69
5.4	Percentage of number of terms in corresponding to from 1 to 10 occurrences in three-level representation.	72
6.1	Original input bitmaps (upper row) and reconstructed output bitmaps (lower row).	78

6.2	DISA model structure for tone selection. C_m stands for the m th Chinese character in a sentence. P_m denotes the pinyin for m th character without the tones. $P_m n$ represents the pinyin for m th character with its n th tone. $F_m n$ is the feature/embedding vector for $P_m n$	83
6.3	An example of fused character feature/embedding lookup, where T, P, V represent features/embeddings from corresponding modality. In the case of single modality or bi-modality, relevant lookup table is constructed accordingly.	86
6.4	The proportion of tokens in testing sets that also appear in training sets. Rows are training sets (T denotes the textual token and P denotes the phonetic token) Columns are testing sets.	95
6.5	Performance comparison between phonetic ablation test groups. rand denotes random generated embeddings. Ex0/Ex04 represent Ex embeddings without/with tones. The same is for PO/PW. + denotes a concatenation operation.	97
6.6	Selected t-SNE visualization of four kinds of phonetic-related embeddings. Circles cluster phonetic similarity. Squares cluster semantic similarity.	100

List of Tables

1.1	Comparison between English and Chinese text in composition.	4
1.2	Examples of intonations that alter meaning and sentiment.	4
2.1	Types of sentiment lexicons.	11
2.2	Comparison between popular Chinese text segmentors	14
3.1	Accuracy of SentiWordNet and SenticNet version (column 2 to 7) and accuracy of small value sentimnet synsets (last 3 columns)	37
3.2	Comparisons between CSenticNet and state-of-art sentiment lexicons .	37
4.1	Comparison with traditional feature on four datasets	49
4.2	Comparison with embedding features on four datasets	49
5.1	Metadata of Chinese dataset	63
5.2	Variants of ATSM-S on Chinese datasets at word level.	66
5.3	Accuracy and Macro-F1 results on Chinese datasets at word level. . . .	68
5.4	Accuracy and Macro-F1 results on single-word/multi-word aspect tar- get subset from SemEval2014.	68
5.5	Accuracy results of multi-granularity with and without fusion mecha- nisms. (W, C, R stands for word, character and radical level, respec- tively. + means a fusion operation.)	71
6.1	Configuration of convAE for visual feature extraction.	78
6.2	Illustration of 4 types of phonetic features: a(x) stands for the extracted audio feature for pinyin ‘x’; v(x) represents learned embedding vector for ‘x’; number 0 to 4 represents 5 diacritics.	80
6.3	Statistics of Chinese characters and ‘Hanyu Pinyin’	81
6.4	Actions in DISA network and meanings.	85
6.5	Statistics of experimental datasets	89
6.6	Classification accuracy of unimodality in LSTM.	91

6.7	Classification accuracy of multimodality. (T and V represent textual and visual, respectively. + means the fusion operation. P is the concatenated phonetic feature of the one extracted from audio (Ex04) and pinyin w/ intonation (PW).)	93
6.8	Cross-domain evaluation. Datasets on the first column are the training sets. Datasets on the first row are the testing sets. The second column represents various baselines and our proposed method.	94
6.9	Performance comparison between learned and random generated phonetic feature.	96
6.10	Performance comparison between different combinations of phonetic features	99

List of Abbreviations

ABSA	Aspect-based Sentiment Analysis
AI	Artificial Intelligence
CBOW	Continuous Bag of Words model
CNN	Convolutional Neural Network
CRF	Conditional Random Field
DNN	Deep Neural Network
LSTM	Long Short-Term Memory
NB	Naive Bayes
NLP	Natural Language Processing
NTUSD	National Taiwan University Sentiment Dictionary
NTU MC	Nanyang Technological University Multilingual Corpus
POS	Part-of-Speech
RNN	Recurrent Neural Network
SOTA	State-of-the-art
SVD	Singular Value Decomposition
SVM	Support Vector Machine

Abstract

Sentiment analysis or opinion mining is a task concerning identifying, extracting and quantifying the sentiment orientations or affective states. The task utilizes a synthesis of techniques like natural language processing, computational linguistics, text mining and so forth. Under its big umbrella, various sub-tasks exist, such as subjectivity detection, sentiment classification, named entity recognition, and sarcasm detection etc. Large quantities of research work that studied the aforementioned tasks were conducted on the English language, due to the popularity of English on the international platform and, thus, its abundance of language resource. Although this research could be applied to other Indo-European languages, they are deficient in performing on many oriental languages, especially on the Chinese language. This was caused by the specific characteristics of the Chinese language.

Inspired by linguistics, this thesis discusses the situations and features that make the Chinese language different from English and proposes corresponding approaches to utilize these opportunities. In the beginning, we reviewed the literature on Chinese sentiment analysis research. Amongst which we noticed that existing Chinese sentiment resource was relatively scarce compared to other languages. This was reflected in two aspects: no semantic connection between words and missing sentiment intensity (fine-grained) measure. Thus, we proposed an unsupervised method to construct a semantic-connected valence Chinese sentiment resource. The mapping-based method leveraged on multiple multilingual and sentiment resources, such as WordNet etc.

Next, we found that Chinese word segmentation could be a source of errors in sentiment analysis, especially in a non-general domain, such as finance or medical. In addition, we analyzed that intra-character components (radicals) of Chinese text carry semantics due to its origin of the pictogram (or ideogram). To this end, we proposed

a radical-based hierarchical character embedding to skip the word segmentation step and also to inject intra-character semantics to the text representation. The new text representation outperformed word-level representation by a considerable margin in the sentiment classification task.

When we tried to extend the hierarchical embedding to aspect-based sentiment analysis task, we realized that existing methods all tend to take the averaged embeddings of multi-word aspect target to represent the aspect target. This assumption will work in English on the condition that the proportion of multi-word aspect target is relatively low. However, almost all Chinese aspect targets are multi-character targets. Thus, we introduced an aspect target sequence modeling (ATSM) network to specifically learn adaptive aspect target representation based on sentence context and ATSM-Fusion network to consider the multi-granularity feature of Chinese text. The ATSM model alone achieved the state-of-the-art performance in English ABSA and ATSM-Fusion pushed the Chinese ABSA performance higher.

In addition to addressing Chinese sentiment analysis from textual modality, we proposed to incorporate phonetic information for textual sentiment analysis. We introduce two effective features to encode phonetic information. Then, we developed a disambiguate intonation for sentiment analysis (DISA) network using a reinforcement network. It functions as disambiguating intonations for each Chinese character (pinyin). Thus, a precise phonetic representation of Chinese is learned. Furthermore, we fused phonetic features with textual and visual features in order to mimic the way humans read and understand Chinese text. Experimental results show that the inclusion of phonetic features significantly and consistently improves the performance of textual and visual representations

In summary, this thesis introduces several approaches to Chinese sentiment analysis, addressing and utilizing the linguistic characteristics (e.g., compositionality, multi-granularity, phonology) that distinguish Chinese from other languages.

Chapter 1

Introduction

1.1 Background

In recent years, sentiment analysis has become increasingly popular for processing social media data on online communities, blogs, wikis, microblogging platforms, and other online collaborative media [2]. Sentiment analysis is a branch of affective computing research [3] that aims to classify text – but sometimes also audio and video [4] – into either positive or negative – but sometimes also neutral [5]. Sentiment analysis is a ‘suitcase’ research problem that requires tackling many NLP sub-tasks, which includes but not limited to subjectivity detection [6], concept extraction [7], aspect extraction [8], sarcasm detection [9], entity recognition [10], personality recognition [11], multimodal fusion [12], user profiling [13] and so forth.

Sentiment analysis techniques can be generally classified into sub-symbolic and symbolic approaches. The sub-symbolic approach includes unsupervised [14], semi-supervised [15] and supervised [16] machine learning-based techniques that leverage on lexical co-occurrence frequencies to classify sentiment. The symbolic approach involves the utilization of resources like lexicons [17], ontologies [18], and semantic networks [19] to infer sentiment from words and multi-word expressions. There are also hybrid approaches [20] that make the most of the advantages from both worlds for sentiment analysis.

Deep neural networks have shown tremendous success in the field of NLP. In the context of sentiment analysis, recursive neural networks [21, 22, 23, 24], convolutional neural networks [25, 26, 27], deep memory networks [28, 29, 30] have achieved

state-of-the-art performance. The attention model was first introduced in image classification. The main purpose of an attention network is to identify and attend the most important representative parts of an object. In the context of NLP, attention networks have recently become extremely popular for machine translation [31, 32], summarization [29], and aspect-based sentiment analysis (ABSA) [30, 33]. While most literature addresses the problem in a language-independent approach, Chinese sentiment analysis, in fact, requires tackling language-dependent challenges due to its unique nature.

Compared to the popular language, English, the Chinese language has the following four major differences. The first difference is that Chinese has a relatively different or even opposite syntactic structure as compared to other languages, especially English, and strategies have to be devised to resolve ambiguities present in Chinese syntactic parsing. For instance, Fig. 1.1 shows how syntax trees differ in English and Chinese of the same sentence.

Figure 1.1: Syntax trees for the sentence “Everything would have been all right if you hadn’t said that” in two languages

The second notable difference in Chinese text is the lack of inter-word spacing because a string of Chinese text is made up of equally spaced graphemes that are called characters. Nevertheless, Chinese does have the concept of word, which consists of various length of character strings. Some words contain only one character (e.g., 他

Unsegmented:	人要是行，干一行行一行，一行行行行。
Correct segmentation:	人 要是 行，干 一行 行 一行，一行 行 行行 行。
Meaning:	People if capable, do one job achieve one job, one job can do all jobs can do.
Incorrect segmentation: <small>(from Stanford parser)</small>	人 要是 行，干 一行行 一行 ，一行 行行 行行。
Meaning:	People if capable, ???

Figure 1.2: Example of importance of word segmentation

(he)), some words contain two characters (e.g., 要是 (if)), some words contain three characters (e.g., 直升机 (helicopter)) and some words contain four characters (e.g., 以德服人 (treat people with morality)). Sentences are the concatenation of these words. Research that works on word-level Chinese sentiment analysis cannot avoid the task of word segmentation. A correct segmentation of words in a sentence will parse correct meanings. On the contrary, an imprecise segmentation will leave the sentence ambiguous and even makes no sense. An example is shown in Fig. 1.2.

The third difference is the property of compositionality of Chinese text. Unlike English, whose fundamental morpheme, such as prefixes, words etc., is a combination of characters, the fundamental morpheme of Chinese is radical, which is a (graphic) component of Chinese characters. Each Chinese character can contain up to five radicals. The radicals within character have various relative positions. For instance, it could be left-right (‘蛤 (toad)’ , ‘秒 (second)’), up-down (‘岗 (hill)’ , ‘孬 (not good)’), inside-out (‘国 (country)’ , ‘问 (ask)’) etc. Their existence is not only decorative but also functional. Radicals have two main functions: pronunciation and meaning. For example, the radical ‘疒’ carries the meaning of disease. Any Chinese character containing this radical is related with disease and, hence, tends to express negative sentiment, such as ‘病 (illness)’ , ‘疯 (madness)’ , ‘瘫 (paralyzed)’ , etc. An example is shown in Table. 1.1.

The fourth difference relates to the phonetics of the Chinese language. Firstly, Chinese spoken system has deeper phonemic orthography than that of other languages, such as English and Japanese, according to the orthographic depth hypothesis [34,

Table 1.1: Comparison between English and Chinese text in composition.

English			Chinese		
Hierarchy	Example	Encodes semantics	Hierarchy	Example	Encodes semantics
Character	a, b, c, ...	N	Radical	彳, 亻, 宀	Y
Character N-gram	pre, sub	partial Y	Character	雪, 林, 伐	Y
Word	awake, cheer	Y	Single-character word	好, 灯	Y
Phrase	kick off, put on	Y	Multi-character word	风景, 大自然	Y
Sentence	Nice to meet you.	Y	Sentence	我很高兴遇见你。	Y

35]. It is hard to support the recognition of words with the language phonology. In comparison for shallow phonemic orthography languages, the pronunciation of a word is largely dependent on the text composition in such languages. One can almost infer the pronunciation of a word given its textual spelling. For instance, if the pronunciations of the English word ‘subject’ and ‘marineland’ were known, it is not hard to speculate the pronunciation of word ‘submarine’, because one can combine the pronunciation of ‘sub’ from ‘subject’ and ‘marine’ from ‘marineland’. As pointed out by Albrow [36], English text is invented to record the pronunciation in its initial time. Whereas Chinese text (written system) belongs to pictogram system, which does not offer pronunciation cues that are as reliable or consistent as those of many other writing systems, such as English.¹ Secondly, as a tonal language, one single syllable in modern Chinese can be pronounced with five different tones, i.e., 4 main tones and 1 neutral tone. This particular form of the Chinese language provides semantic cues complementary to its textual form as illustrated in Table 1.2.

Table 1.2: Examples of intonations that alter meaning and sentiment.

Text	Pronunciation	Meaning	Sentiment polarity
空	kōng	Empty	Neutral
	kòng	Free	Neutral
假	jiǎ	Fake	Neutral/Negative
	jià	Holiday	Neutral
好吃	hǎochi	Delicious	Positive
	hàochi	Gluttony	Negative

¹Although phonograms (or phono-semantic compounds, xingsheng zi) are quite common in the Chinese language, only less than 5% of them have exactly the same pronunciation and intonation. <https://zhuanlan.zhihu.com/p/38129192>

1.2 Challenges

There are currently numerous English-language sentiment knowledge bases already in existence, such as SenticNet [37] and SentiWordNet [38]. When it comes to the Chinese language, however, the numbers of similar resources are insufficient. Two major sentiment lexicons are currently available in Chinese: HowNet [39] and NTUSD [40]. However, both have their own drawbacks: HowNet only provides a positive or negative label for words. The labeling polarity does not give users information as to what extent a word expresses a sentiment. For example, *uneasy* and *indignant* are both negative-connotation words but to different extents. HowNet classified these two words as equals in the ‘negative’ list with no discrepancy between them. The entries in HowNet are basically simple words or idioms. As the fundamental elements (word level) in Chinese sentences and passages, their contribution to the overall sentiment is trivial compared with multi-word phrases. Furthermore, HowNet lacks semantic connections between its words. Their words are simply listed in pronunciation order, which makes it impossible to infer sentiment from semantics. Although bigger than HowNet in size, NTUSD contains all the above drawbacks. To conclude, they are all word-level polarity lexicons that lack fine-grained sentiment score and semantic inference capability.

In terms of the composition property of Chinese characters, [41] started to break down Chinese words to characters. They proposed a character-enhanced word embedding model (CWE). [42] started to break down Chinese characters to Chinese radicals (pianpang bushou). They developed a radical-enhanced Chinese character embedding. In their method, only one radical from each Chinese character was selected to assist the character representation. [43] began to train pure radical-based embedding for web search ranking, Chinese word segmentation, and short-text categorization. Yin et al. [44] introduced multi-granular Chinese word embeddings to improve the pure radical embedding. Nevertheless, no literature has decoded the semantics in Chinese radicals for sentiment analysis task or designed radical-based representations for Chinese sentiment analysis.

ABSA is a fine-grained sentiment analysis task that received massive attention in the community due to its wide application scenario in real life. ABSA contains multiple subtasks, such as aspect term extraction, aspect term classification, aspect category classification and so forth. Among them, aspect term classification is extremely popular. Aspect term is sometimes also termed as aspect target, depending on the context. In aspect target sentiment classification, Tang et al. [45] used a bidirectional-LSTM model to encode sentence and aspect target sequential information. They then concatenated the target embedding to each sentence word embedding to emphasize the interplay between target and sentence contexts. In [30], they introduced an attention-based memory network to particularly model the correlation between aspect target and sentence contexts. Wang et al. [33] also came up with an attention-based network on top of a sentence level LSTM encoder to learn aspect target representation.

The above literature shared some common disadvantages. Firstly, they treat aspect target as a helper to find the sentence sentiment, whereas it should be the opposite. Sentiment should be expressed towards aspect target itself, instead of sentence. For example, “The hotel room is small, but the view is nice.” has a positive sentiment towards “view” and a negative sentiment towards “room”. Secondly, general embeddings were used to represent aspect target which will result in ambiguity. For instance in the sentence “I’m so happy to buy the red apple of 64G but not 32G.”, “red apple” is no longer a fruit but a symbol for iPhone. If general embeddings of fruit apple are used here, ambiguity will emerge and will mislead the sentiment classification. Thirdly, the state of the arts all oversimplify multi-word aspect target by taking the averaged embeddings. Sequential information within aspect target is lost and aspect target embedding ends up in an unknown position in the word vector space.

Various Chinese sentiment analysis approaches have been actively explored in the past few year, from document level [46, 47], to sentence level [25, 48] and to aspect level [49, 50]. Most methods took a high perspective to develop effective models for a broad spectrum of languages. Only a limited number of works spend efforts in studying language-specific characteristics [51, 52, 53]. Among them, there is almost

Figure 1.3: Research path and motivation

no literature trying to take advantage of phonetic information for Chinese representation. We, however, believe the Chinese phonetic information could of great value to the representation and sentiment analysis of the Chinese language, due to its deep phonemic orthography and intonation variation.

1.3 Contributions

In order to address the issues exposed in Section 1.2, this thesis presents the following pieces of work which each focuses on one aspect of Chinese sentiment analysis. An overview of the research path of this thesis is given in Fig. 1.3.

- We present a method to construct the first fine-grained concept-level Chinese sentiment resource with semantic correlation. The method defines mapping algorithms between multiple English and Chinese lexical resources to automate the construction of new Chinese resource. Different techniques were proposed to tackle issues like sense ambiguity and non-exact match. Two types of Chinese sentiment resources were introduced, depending on the English resource used (SentiWordNet and SenticNet). The SenticNet version achieves the state-of-the-art performance in the evaluations.
- We propose Chinese radical-based hierarchical embeddings particularly designed for sentiment analysis. Four types of radical-based embeddings were introduced:

radical semantic embedding, radical sentic embedding, hierarchical semantic embedding, and hierarchical sentic embedding. By conducting sentence-level sentiment classification experiments on four Chinese datasets, we proved the proposed embeddings outperform state-of-the-art textual and embedding features.

- We investigate the problem of ABSA from a new perspective where the aspect target sequence dominates sentiment classification. Accordingly, we propose an ATSM-S model, which outperforms the state of the art in multi-word aspect sentiment analysis on SemEval2014 dataset. Furthermore, we extend the ATSM-S to ATSM-F to consider the multi-granularity property of Chinese text by fusing the representations from radical, character and word level. The ATSM-F model prevails all state of the arts in Chinese review sentiment analysis.
- We present an approach to learn phonetic information out of pinyin (both from audio clips and pinyin token corpus) and design a network to disambiguate intonations. With the learned phonetic information, we integrate it with textual and visual features to create new Chinese representations. Experiments on five datasets demonstrated the positive contribution of phonetic information to Chinese sentiment analysis.

1.4 Organization

The rest of this thesis is organized below. Chapter 2 presents a literature review on recent progress in Chinese sentiment analysis; Chapter 3 introduces a concept-level Chinese sentiment resource, CsenticNet. Next, two characteristics of the Chinese language were studied. Chapter 4 studies the characteristic of compositionality by describing approaches to consider radical components for Chinese sentiment analysis; Chapter 5 extends the radical-enhanced embedding to aspect-based sentiment analysis; Chapter 6 studies the characteristic of phonology by incorporating phonetic information to Chinese text representation for sentiment analysis; Finally, Chapter 7 concludes the thesis and proposes future work.

Chapter 2

Literature Review

Although there is plenty of research on English sentiment analysis for the last decade, relevant research in the Chinese language is limited. Only in recent years did researchers begin to conduct sentiment analysis in the Chinese language [54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82].

This chapter presents an overview of the existing works for Chinese sentiment analysis. It first introduces existing Chinese sentiment resources. Next, monolingual approaches and multilingual approaches were presented, respectively. Then, text representation and phonology that relate to the characteristics of the Chinese language were reviewed.

2.1 Sentiment Resource

The sentiment resource can be generally classified into two categories, corpus and lexicon. They are two kinds of different language structures, where each induces its own sentiment classification methods.

2.1.1 Corpus

A corpus is a collection of complete and self-contained texts, e.g., the corpus of Anglo-Saxon verses [83]. A corpus, in linguistics and lexicography, is defined as a body of texts, utterances, or other specimens considered more or less representative of a language, which usually rendered in a machine-readable format. Corpora can store

millions of words with features that could be extracted by means such as tagging, parsing and the use of concordant programs.

The corpus usually contains plenty of semantics made of emotional expressions in textual forms, such as words, phrases, sentences, and paragraphs. It is the fundamental material for building an emotion system due to its richness in semantics. Nevertheless, the lack of large and annotated Chinese sentiment corpora has hindered the progress of Chinese sentiment analysis. To this end, one branch of researchers attempted to expand or modify the existing Chinese sentiment corpus.

Multi-granular (from the sentence to paragraph and document) emotion annotation scheme was proposed by [54]. Specifically, the corpus was annotated by eight emotions categories (hate, anxiety, angry, sorrow, joy, expectant, surprise and love). The downside of the method was the laborious and time-consuming manual annotation. Instead of using manual work, [84] introduced a Chinese Sentiment Treebank of social data, where they crawled over 13K movie review sentences online. Afterward, they developed a recursive neural deep model (RNDM) to assign sentiment labels towards each sentence.

Constructing a corpus is important but definitely not the final goal. Analyzing the developed corpus is an essential work [55]. Many existing English sentiment corpora had sentiment details only at sentence level[85, 86, 87]. Zhao et al., however, built a global fine-grained corpus whose annotation introduced cross-sentence and global emotion information. They also introduced target-aspect pair extraction and implicit polarity extraction tasks in the work. On top of mono-lingual approach, Lee et al. [88] extended to multi-lingual corpus by collecting and annotating a code-switching (more than one language occur in the same sentence) emotion corpus of English and Chinese from Weibo. The corpus is generally used by machine learning methods. Conventional methods extract features of different kinds, such as syntactic, semantic, and lexical features, and feed these features to a classifier for training and testing. Since the 2010s, deep neural networks (DNN) have been dominating over standard machine learning classifiers (such as support vector machine, naive bayes etc.) One advantage

Table 2.1: Types of sentiment lexicons.

Type	Characteristic	Example
1	Contain sentiment words	NELL[90]
2	Contain sentiment words + polarity	NTUSD[40], HowNet[39]
3	Contain sentiment words+ polarity+intensity	SentiWordNet[38], SenticNet[37]

of DNN is its ability to learn feature representation dynamically and automatically from the corpus, which saves the researcher from laborious feature engineering.

2.1.2 Lexicon

Compared with the corpus, the lexicon offers obvious clues to sentiment analysis. Sentiment/emotion lexicon consists of words or phrases that directly express subjective emotions, sentiment or opinions [89]. It is vital to sentiment analysis because it provides a ground-truth reference of sentiment polarity. Sentiment lexicons can be classified into three types based on what kind of information they provide [56], as shown in Table 2.1.

In the literature, there are two approaches to build a sentiment lexicon. The first approach involves lots of human labor. Language practitioners will need to collect sentiment words and phrases and annotate them with sentiment polarity labels manually. This method is not only restricted by the knowledge of human experts but also lacks the possibility for future expansion. Thus, it is mostly utilized for experimental evaluation in the end of automatic algorithms.

The second approach relies on lexical dictionaries. A dictionary is a lexical resource that usually organizes words or phrases by spelling (for English) or pronunciation (for Chinese). Each word was paraphrased by meanings, antonyms, and synonyms. These explaining contents link various words together and provide a path to refer to each other. The second approach usually starts with a seed list, which contains a small set of sentiment words. By looking up their semantic neighborhood (such as synonyms or antonyms) in the dictionary, new sentiment words can be added to the initial seed list. The process will iteratively be applied to the newly collected words. The original seed list will gradually grow larger. One typical example of this kind is the HowNet [91]. It

is a bilingual (English and Chinese) commonsense knowledge base. Chinese sentiment lexicon is a subset of HowNet. Many researchers started from this lexicon and used it in multiple improved ways.

Liu et al. [58] proposed a framework to improve domain-specific sentiment classification performance via a domain-specific sentiment lexicon. A constructed lexicon is combined with existing sentiment lexicons, like HowNet to apply on aspect review opinion mining. Xu et al. [59] modified and expanded HowNet and NTUSD [40] with a large unlabeled corpus from Sogou. Rather than improving the traditional lexicons, Xu et al. [60] came up with a graph-based method with the help of a few seed words. They extracted the common features between words and multi-resources to improve the method iteratively. In the end, human experts double examined the lexicon manually. Wu et al. [56] introduced “iSentiDictionary”, a Chinese sentiment lexicon from semantic graph. They extracted a list of seed words from traditional sentiment lexicons and categorize them to four classes. Seed words from each class were fed to a self-training spreading algorithm to retrieve more relevant words on ConceptNet. The retrieved words were added to the original seed word list.

Wang et al. [61] pointed out that existing methods failed to consider the fuzziness of sentiment polarity. They argued that sentiment words might carry opposite sentiment orientations in a different context. To address this issue, they developed a fuzzy computing model (FCM) to detect sentiment polarity of a word, which includes three modules, a key sentiment morpheme set, sentiment words datasets, and a key sentiment lexicon. They firstly obtained polarity scores from each of the module. Then, they trained a fuzzy classifier to adjust the parameters. Thus a dynamical sentiment polarity will be learned. Their method has achieved state-of-the-art performance.

Since the resource of English is more abundant than Chinese, some researchers start to leverage on bilingual resources. Su and Li [57] introduced a bilingual method to build Chinese sentiment lexicon. They obtained the sentiment orientations of Chinese words by computing the PMI value of English seed words, where the English and Chinese words were mutually translated. Gao et al. [92] modeled the lexicon learning as a bilingual word graph. The graph comprised of two layers, one for each language.

Sentiment words in each language/layer were connected by a positive weight with their synonyms and negative weight with their antonyms. The two layers were then linked by an inter-language sub-graph. Through a word graph-label propagation algorithm, sentiment orientations of Chinese words were inferred from the labeled English seed words.

2.2 Monolingual Approach

We first present Chinese word segmentation tools in this section. Then, we introduce sentiment classification methods using machine learning-based methods and then the knowledge-based approach.

Figure 2.1: Hourglass model in the Chinese language

2.2.1 Preprocessing

The first step towards Chinese textual processing is usually word segmentation. Three most popular Chinese word segmentors are Jieba segmentor, THULAC, and ICT-

CLAS. Jieba segmentor¹ is an open source segmentor with up to 9 programming language adaptation. THULAC [93] was developed by Tsinghua university with decent balance between speed and accuracy in performance. ICTCLAS [71] was invented by Dr. Zhang which has the best performance. A comparison among these segmentors was shown in Table 2.2². After word segmentation, fundamental textual preprocessings like tokenization and POS tagging can be conducted by common tools, such as Scikit-learn [94], NLTK [95] and so forth.

Table 2.2: Comparison between popular Chinese text segmentors

Algorithm	F-Measure		Speed	Supported Language
	msr_test (560KB)	pku_test (510KB)	CNKLjournal.txt (51MB)	
ICTCLAS(2015)	0.891	0.941	490.59KB/s	C, C++, C#, Java, Python
Jieba(C++)	0.811	0.816	2314.89KB/s	C++, Java, Python, R, etc.
THULAC_lite	0.888	0.926	1221.05KB/s	C++, Java, Python, SO

2.2.2 Machine Learning-based Approach

The machine learning approach usually refers to supervised learning which models the sentiment analysis as a text classification task. It requires a labeled dataset but not any specific predefined semantic rules. The general steps to supervised learning start with the extraction of textual features, such as lexical features (n-grams), syntactic features (POS tags), semantic features (semantic graph path) and so forth. Next, a machine learning classifier is trained and tested. The process is shown in Fig. 2.2. In addition to the conventional machine learning framework, deep neural networks save researchers from laborious feature engineering.

Figure 2.2: Machine learning-based processing for Chinese sentiment analysis

Historically, research direction could be classified into three major groups based on the different procedures used in machine learning sentiment classification. The

¹<https://github.com/fxsjy/jieba>

²Data source: <http://thulac.thunlp.org/>

first group focused on studying features. In addition to the commonly employed n-gram feature, Zhai et al. pointed out [63] that seldom used structures like sentiment words, substrings, substring groups, and key substring groups can also be used to extract features. Their analysis suggested that different types of features have different capabilities of discrimination, and substring-group features may have the potential for better performance. Su et al. [64] tried making use of semantic features and applied word2vec, which utilized neural network models to learn vector representations of words. After the extraction of deep semantic relations (features), word2vec is used to learn the vector representations of candidate features. The authors finally applied SVM as a classification technique on their features and achieved an accuracy level of over 90%.

Xiang [65] presented a novel algorithm based on an ideogram. The method does not need a corpus to compute a word's sentiment orientation but instead requires the word itself and a pre-computed character ontology (a set of characters annotated with sentiment information). The results revealed that their proposed approach outperforms existing ideogram-based algorithms. Some researchers developed novel neural features by combining the compositional characteristic of Chinese with deep learning methods. Chen et al. [41] started to decompose Chinese words into characters and proposed a character-enhanced word embedding model (CWE). Sun et al. [42] started to decompose Chinese characters to radicals and developed a radical-enhanced Chinese character embedding. However, they only selected one radical from each character to enhance the embedding.

Shi et al. [43] began to train pure radical-based embedding for short-text categorization, Chinese word segmentation, and web search ranking. Yin et al. [44] extend the pure radical embedding by introducing multi-granularity Chinese word embeddings. However, none of the above embeddings have considered incorporating sentiment information and apply the radical embeddings to the task of sentiment classification.

As compared to the first group, however, the second group that focuses on studying different classification model is more popular. Xu et al. [66] proposed an ensemble

learning algorithm based on a random feature space-division method at document level, or multiple probabilistic reasoning model (M-PRM). The algorithm captures and makes full use of discriminative sentiment features. Li et al. [84] introduced a novel recursive neural deep model (RNDM) which can predict sentiment labels based on recursive deep learning. It is a model that focuses on sentence-level binary sentiment classification and claims to outperform NB and SVM. Cao et al. introduced a joint model which incorporated SVM and a deep neural network [67]. They considered sentiment analysis as a three-class classification problem, and designed two parallel classifiers before merging the two classifiers' results as the final output. The first classifier was a word-based vector space model, in which unlabeled data was firstly identified and then added to a sentiment lexicon. Features were then extracted from the sentiment lexicon and labeled training data.

Before building the SVM classifier, training data was then processed in order to make it more balanced. The second classifier was an SVM model, in which distributed paragraph representation features were learned from a deep convolutional neural network. Finally, the two classifiers' results were merged with an emphasis on neutral output, or, the second classifier's output. Liu et al. used a self-adaptive hidden Markov model (HMM) in [68] to conduct emotion classification. They used Ekman's [96] six well-known basic emotion categories: happiness, sadness, fear, anger, disgust, and surprise. Initially, they designed a category-based feature. For each emotion category, they computed the MI, CHI, term frequency-inverse document frequency (TF-IDF) and expected cross entropy (ECE). The four results formed the four dimensions of the category-based feature. Then, a modified HMM-based emotional classification model was built, featuring a self-adaptive capability through the use of a particle swarm optimization (PSO) algorithm to compute the parameters. The model performed better than SVM and NB in certain emotion categories.

The third group has attempted to develop new machine learning-based approaches in Chinese sentiment classification. Wei et al. [97] presented a clustering-based Chinese sentiment classification method. Sentiment sequences were first built from micro-blogs such as Weibo, and Longest Common Sequence algorithms were then applied

to compute the differences between two sentiment sequences. Following, a k -medoids clustering algorithm was applied at the end of the process. This method does not require a training data and yet provides efficient and good performance for short Chinese texts.

Ku et al. [69] applied morphological structures and relations between sentence segments to Chinese sentiment classification. CRF and SVM classifiers were used in the model and results indicate that structural trios benefit sentence sentiment classification. Xiong [70] developed an ADN-scoring method using appraisers, degree adverbs, negations, and their combinations for sentiment classification of Chinese sentences. A particle swarm optimization (PSO) algorithm was also used to optimize the parameters of the rules for the method.

Chen et al. proposed a joint fine-grained sentiment analysis framework at the sub-sentence level with Markov logic in [41]. Unlike other sentiment analysis frameworks where subjectivity detection and polarity classification were employed in sequential order, Chen et al. separated subjectivity detection and polarity classification as isolated stages. The two separated stages were learned by local formulas in Markov logic using different feature sets, like context POS or sentiment scores. Next, they were integrated into a complete network by global formulas. This, in turn, prevented errors from propagating due to chain reactions.

In addition to the classical binary or ternary label classification problem (positive, negative or neutral), multi-label classification research has also recently gained popularity. Liu et al. [98] proposed a multi-label sentiment analysis prototype for micro-blogs and also compared the performance of 11 state-of-the-art multi-label classification methods (BR, CC, CLR, HOMER, RAKEL, ECC, MLkNN, RF-PCT, BRkNN, BRkNN-a and BRkNN-b) on two micro-blog datasets. The prototype contained three main components: text segmentation, feature extraction and multi-label classification. Text segmentation segmented a text into meaningful units. Feature extraction extracted both sentiment features and raw segmented words features and represented them in a bag of words form. Multi-label classification compared all 11

methods' classification performance. Detailed experimental results suggested that no single model outperformed others in all of the test cases.

2.2.3 Knowledge-based Approach

Another popular approach is the knowledge-based approach or usually termed the unsupervised approach. After the preprocessing of texts, knowledge-based approach divides into two branches. The first branch relies on a sentiment lexicon to find the sentiment polarity of each phrase obtained in the step before. It then sums up the polarities of all phrases in sentences, paragraphs or documents (based on required granularity). If the summation is greater than zero, the sentiment of this granularity will then be positive, and vice versa if the summation is less than zero. The second branch tries to explore syntactic rules and other logic.

For instance, semantic orientation (SO) is estimated from the extracted phrases using the Point Wise Index algorithm. Eventually, the average SO of all phrases is computed. If the average value is greater than zero, the sentiment of the phrases in the document is classified as positive, and vice versa if the average value is less than zero. Researchers tend to prefer the second branch due to the greater flexibility offered. Zhang et al.[72] proposed a rule-based approach with two phases: the sentiment of each sentence is first decided based on word dependency; and, aggregating the sentences' sentiments then generates the sentiment of each document.

Zagibalov et al. [73] presented a method that does not require any annotated training data and only requires information on commonly occurring negations and adverbials. The performance of their method is close to and sometimes outperforms supervised classifiers. Recent research [74] considers both positive/ negative sentiment and subjectivity/objectivity as a continuum. The unsupervised techniques were used to determine the opinions present in the document. The techniques include a one-word seed vocabulary, iterative retaining for sentiment processing and a criterion of "sentiment density". Due to lexical ambiguity of the Chinese language, much work has been conducted on fuzzy semantics in Chinese.

Li et al. [75] claimed that polarities and strengths of sentiment words obey a Gaussian distribution, and proposed a Normal distribution-based sentiment computation method for quantitative analysis of semantic fuzziness of Chinese sentiment words. Zhuo et al. [76] presented a novel approach based on the fuzzy semantic model by using an emotion degree lexicon and a fuzzy semantic model. Their model includes text preprocessing, syntactic analysis and emotion word processing. Optimal results of their model are achieved when the task is clearly defined. The unsupervised approach can also be applied to aspect-level sentiment classification. Su et al. [77] presented a mutual reinforcement approach to study the aspect-level sentiment classification. An attempt to simultaneously and iteratively cluster product aspects and sentiment words was performed. The authors constructed an association set of the strongest n-sentiment links, which was used to exploit hidden sentiment association in reviews.

Some recent research work also studied the discourse and dependency relations of Chinese data. In [78], Wu et al. studied the combination problem of Chinese opinion elements. Opinion topics (topic, feature, item, opinion word) were extracted from documents based on lexicons. Features were then combined with three sentence patterns (general sentences, equative sentences and comparative sentences) in order to predict the opinion. These sentence patterns determined how the opinion elements in the sentence should be computed to yield the sentiment of a whole sentence. Quan et al. went further in sentiment classification using dependency parsing in [99]. They integrated a sentiment lexicon with dependency parsing to develop a sentiment analysis system. They firstly conducted a dependency analysis (*nsubj*, *nn*, *advmod*, *punct*) of sentences so as to extract emotional words. Based on these, they established sentiment dictionaries from a lexicon (HowNet) and calculated word similarities to predict the sentiment of sentences.

So far, we have seen that Chinese sentiment analysis research has restricted elementary components to word or character level. Even though state-of-the-art algorithms (either machine learning-based or knowledge-based) performed reasonably well, word level analysis did not reflect real human reasoning faithfully. Instead, concept level reasoning needs to be explored as it has been demonstrated to be closer

Figure 2.3: Evolution of NLP research through three different eras from [1]

to the truth [20]. Our mental world is a relational graph whose nodes are various concepts. As Fig. 2.3 from [1] shows, NLP research is gradually shifting from lexical semantics to compositional semantics. To the best of our knowledge, there is no current work on concept level Chinese sentiment analysis. Thus, related works in the future are expected to be promising.

2.2.4 Mix Models

Finally, there is also a branch of researchers who have combined the machine learning approach with a knowledge-based approach. Zhang et al. [79] introduced a variant of the self-training algorithm, named EST. The algorithm integrates a lexicon-based approach with a corpus-based approach and assigns an agreement strategy to choose new, reliable labeled data. The authors then proposed a lexicon-based partitioning technique to split the test corpus and embed EST into the framework. Yuan et al. [80] conducted Chinese micro-blog sentiment classification using two approaches. For the unsupervised approach, they integrated a simple sentiment word-count method (SS-WCM) with three Chinese sentiment lexicons. For the supervised approach, they tested three models (NB Classifier, Maximum Entropy Classifier and Random Forests Classifier) with multiple features.

Their results indicated that the Random Forests Classifier provided the best performance among the three models. Li et al. [100] presented a model that boasted the combination of a lexicon-based classifier and a statistical machine learning-based

classifier. The output of the lexicon-based classifier was simply the sum of the sentiment scores of each word in the sentence. For the machine learning-based classifier, they selected unigram and weibo-based features from many candidate features so as to train an SVM classifier. Finally, the system gave a linear combination of the two classifiers' outputs. Likewise, in [101], Wen et al. introduced a method based on class sequential rules for emotion classification of micro-blog texts. They firstly obtained two emotion labels of sentences from lexicon-based and SVM-based methods. Then, the conjunctions of previous emotion labels sequences were formed and class sequential rules (CSR) were mined from the sequences. Eventually, features were extracted from CSRs and a corresponding SVM classifier was trained to classify the whole text.

2.3 Multilingual Approach

Natural language processing research in English language dates back to the 1950s [1]. Within this general multi-disciplinary field, English sentiment analysis only developed about twenty years ago. In comparison, Chinese sentiment classification is a relatively new field with a history only dating back about ten years. As such, resources available for sentiment classification in English are more abundant than those in Chinese, and some researchers have therefore taken advantage of the established English sentiment classification structure to conduct research on Chinese sentiment classification.

Wan [81] proposed a method that incorporates an ensemble of English and Chinese classifications. The Chinese language reviews were first translated into English via machine translation. An analysis of both English and Chinese reviews was then conducted and their results combined to improve the overall performance of the sentiment classification. The problem of the above-mentioned approach is that the output of machine translation is unreliable if the domain knowledge is different for the two languages. This could lead to an accumulation of errors and reduce the accuracy of the translation. As a result, some researchers formulate this into a domain adaptation. Wei and Pal [102] showed that, rather than using automatic translation, the application of techniques like structural correspondence learning (SCL) that link two

languages at the feature-level can greatly reduce translation errors. Additionally, He et al. [82] proposed a method that does not need domain and data specific parameter knowledge.

Language-specific lexical knowledge from available English sentiment lexicons is incorporated through machine translation into latent Dirichlet allocation (LDA) where sentiment labels are considered topics [103]. This process does not introduce noise to the process due to the high accuracy of the lexicon translation and does not need labeled corpus for training. Finally, instead of solely improving the performance of the Chinese language analysis, Lu et al. [104] developed a method that could jointly improve the sentiment classification of both languages. Their approach involves an expectation maximization algorithm that is based on maximum entropy. It jointly trains two sentiment classifiers (each language a classifier) by treating the sentiment labels in the unlabeled parallel text as unobserved latent variables. Together with the inferred labels of the parallel text, the joint likelihood of the language-specific labeled data will then be regularized and maximized. Zhou et al. [105] incorporated sentiment information into Chinese-English bilingual word embeddings using their proposed denoising autoencoder. Chen et al. [106] introduced a semi-supervised boosting model to reduce the transferred errors of cross-lingual sentiment analysis between English and Chinese.

2.4 Text Representation

2.4.1 General Embedding Methods

One-hot representation is the initial numeric word representation method in NLP [107]. However, it usually leads to a problem of high dimensionality and sparsity. To solve this problem, distributed representation or word embedding was proposed [108]. Word embedding is a representation which maps words into low dimensional vectors of real numbers by using neural networks. The key idea is based on the distributional hypothesis so as to model how to represent context words and the relation between context words and the target word. Thus, the language model is a natural solution. Bengio et al. [109] introduced neural network language model (NNLM) in 2001.

Instead of using counts to model ngram language model, they built a neural network. Word embeddings are the byproducts of building the language model. In 2007, Mnih and Hinton proposed a log-bilinear language model (LBL) [110] which is built upon NNLM and later upgraded to hierarchical LBL (HLBL) [111] and inverse vector LBL (ivLBL) [112]. Instead of modeling ngram model like the above, Mikolov et al. [113] proposed a model based on recurrent neural networks to directly estimate the probability of target words given contexts.

Since the introduction of the C&W model [114] in 2008, people started to design models whose objectives are no longer the language model but the word embedding itself. C&W places the target word in the input layer, and output only one node which denotes the likelihood of the input words' sequence. Later in 2013, Mikolov et al. [115] introduced the continuous bag-of-words model (CBOW), which places context words in the input layer and target word in the output layer, and Skip-gram model, which swaps the input and output in CBOW. They also proposed negative sampling which greatly speeds up training. In 2014, Pennington et al. [115] created the 'GloVe' embeddings. Unlike the previous which learned the embeddings from minimizing the prediction loss, GloVe learned the embeddings with dimension reduction techniques on co-occurrence counts matrix.

2.4.2 Chinese Text Representation

Chinese text differs from English text for two key aspects: it does not have word segmentations and it has a characteristic of compositionality due to its pictogram nature. Based on the former aspect, contemporary Chinese text processing mostly relies on Chinese word embeddings [116, 117]. Word segmentation tools are always employed before text representation, such as ICTCLAS [71], THULAC [118], Jieba³ and so forth. Based on the latter aspect, however, the Chinese word consists of sub-elements like characters that contain semantics. Several works had focused on the use of sub-word components (such as characters and radicals) to improve word embeddings. Xu et al. [119] studied characters within a word can enrich semantics for

³github.com/fxsjy/jieba

Chinese word and character embeddings. Chinese text has one level below character level, which is radical level. It has been demonstrated that radical level representation also encodes certain extent of semantics [43]. Chen et al. [41] started to decompose Chinese words into characters and proposed a character-enhanced word embedding model (CWE). Sun et al. [42] started to decompose Chinese characters to radicals and developed a radical-enhanced Chinese character embedding. Shi et al. [43] began to train pure radical-based embedding for short-text categorization, Chinese word segmentation, and web search ranking. Li et al. [120] proposed component-enhanced Chinese character embedding models by incorporating internal compositions and the external contexts of Chinese characters. Yin et al. [44] extended the pure radical embedding in [121] by introducing multi-granularity Chinese word embeddings. Multi-modal representation in the past few years has become a growing area of research. [122] and [52] explored integrating visual features to textual word embeddings. The extracted visual features proved to be effective in modeling the compositionality of Chinese characters.

However, none of the above embeddings have considered incorporating sentiment information and apply the radical embeddings to the task of sentiment classification. Few literature has exploited the multi-grained characteristic of Chinese text in complex NLP problems, such as ABSA.

2.5 Chinese Phonology

Most methods took a high perspective to develop effective models for a broad spectrum of languages. Only a limited number of works spend efforts in studying language-specific characteristics [51, 52, 53]. Among them, there is almost no literature trying to take advantage of phonetic information for Chinese representation. We, however, believe the Chinese phonetic information could of great value to the representation and sentiment analysis of the Chinese language, due to but not limited to the following evidence.

Shu and Anderson conducted a study on Chinese phonetic awareness in [123]. The study involved 113 participants of Chinese 2nd, 4th, and 6th graders enrolled

in a working-class Beijing, China elementary school. Their task was to represent the pronunciation of 60 semantic phonetic compound characters. Results showed that children as young as 2nd graders are better able to represent the pronunciation of regular characters than irregular characters or characters with bound phonetics. The strong influence of familiarity on pronunciation underlines an unavoidable fact about the Chinese writing system: the system does not offer pronunciation cues that are as reliable or consistent as those of many other writing systems, such as English [36]. Moreover, Hsiao and Shillcock argued that semantic-phonetic compound (or phonetic compound) comprised about 81% of the 7000 frequent Chinese characters [124]. These compounds would affect semantics greatly if we can find an approach to effectively represent their phonetic information.

No previous work has integrated pronunciation information to Chinese representation. Due to its deep phonemic orthography, we believe the Chinese pronunciation information could elevate the representations to a higher level.

2.6 Summary

This chapter reviewed recent progress in Chinese sentiment analysis, which includes sentiment resource, monolingual/multilingual approaches, Chinese text representation and Chinese phonology. We realized that current research on Chinese sentiment analysis seldom considers the study of concept-level knowledge in texts. The compositionality characteristic of Chinese text is not thoroughly explored in coarse or fine-grained sentiment analysis task. Moreover, the Chinese phonology which could play an important role in Chinese text representation lacks investigation.

Chapter 3

CsenticNet

3.1 Introduction

Nowadays, sentiment knowledge bases of English are quite affluent, such as SenticNet [37] and SentiWordNet [38]. Whereas for the Chinese language, there is insufficient sentiment resource, especially with the sentiment lexicons. Although HowNet [39] and NTUSD [40] are the most popular two lexicons, they have the following drawbacks. HowNet contains one list of positive words and one list of negative words. Except for the sentiment polarity, no sentiment arousal information can be extracted from the words in the lists. This made the fine-grained sentiment analysis task hard to accomplish. For instance, ‘ecstasy’ and ‘pleasant’ are two words having the same sentiment polarity but different intensity. Only knowing sentiment polarity will not recognize that ‘ecstasy’ carries more positive sentiment than ‘pleasant’. Moreover, HowNet lacks multi-word expressions which limits the number of items that can be matched in the lexicon. In addition, sentiment words in the lexicon are ordered based on pronunciation which makes it impossible to infer sentiment from semantics. NTUSD has all the above shortcomings, although with a larger size.

To conclude, they are all word-level polarity lexicons that lack fine-grained sentiment score and semantic inference capability. Because of these problems in the existing lexicons, we propose a method to construct a concept-level sentiment resource in simplified Chinese to tackle the above issues, taking advantage of existing English sentiment resources and multi-lingual corpus.

3.2 Background

There are basically three types of sentiment lexicons in all [56]: 1) The ones only containing sentiment words, such as The never-ending language learner (NELL) [90]; 2) The ones containing both sentiment words and sentiment polarities (sentiment orientation), such as National Taiwan University sentiment dictionary (NTUSD) [40] and HowNet [39]; 3) The ones containing words and relevant sentiment polarity values (sentiment orientation and degree), such as SentiWordNet [38] and SenticNet [37]. In the first type, the lexicon only contains words for certain sentiments. It can help distinguish texts with sentiments from those that without. However, it is not able to tell whether the texts have positive or negative sentiments. Furthermore, it is an English language corpus and not Chinese sentiment-related. The second, HowNet [39], is an on-line common-sense knowledge base which represents concepts in a connected graph. In terms of its sentiment resources, it has two lists which sentiment words are classified under: positive and negative. The problem this poses is a three-fold one. Firstly, it lacks semantic relationship among the words, as words are listed in alphabetical order. Secondly, it lacks multi-word phrases. Thirdly, it cannot distinguish the extent of the sentiment expressed by the words. For example, *uneasy* and *indignant* are both negative-connotation words but to different extents. HowNet classified these two words as equals in the ‘negative’ list with no discrepancy between them. NTUSD also has the above disadvantages.

With regards to the third type, both SentiWordNet and SenticNet provide polarity values for each entry in the lexicon. They are currently the most state-of-the-art sentiment resources available. However, their drawback is that they are only available in the English language and, hence, do not support the Chinese language sentiment analysis. Thus, some researchers seek to build sentiment resources via a multi-lingual approach. Mihalcea et al. [125] tried projections between languages, but they have the problem of sense ambiguity during translation and time-consuming annotation.

To this end, we introduce an approach to utilize multi-lingual resource to build a Chinese sentiment resource (which contains sentiment words and relevant sentiment

Figure 3.1: CSenticNet

polarity values). The approach extracts the latent connection between two resources to map the English entity to Chinese in a dedicated way. Particularly, the issue of sense ambiguity and non-exact match problem were tackled. Compared to the existing multi-lingual approach [126, 106, 127, 128], no machine translation or mapping learning step is involved.

3.3 Overview

We firstly introduce the multi-lingual resources used, and then briefly present two versions of mapping. The final output of CSenticNet includes a list of concept nodes (synsets), where each node contains a sub-list comprised of words or phrases of similar semantics. Each node also has one sentiment polarity value that is shared between the concept and semantics. An illustration is shown in Fig. 3.1(a).

3.3.1 Resources

Several resources were utilized in our method, such as *SenticNet* [37], *Princeton WordNet* [129], *NTU multi-lingual corpus* [130] and *SentiWordNet* [38]. We present them in details below.

SenticNet [37] is an English resource comprised of concept nodes. Under each of the 17k concept node, there rest 5 affiliated nodes which share similar semantics. These concept and semantic nodes were either multi-word expressions or single words, shown in Fig. 3.2. In addition, each concept node was represented by 4 sentics and 1

Figure 3.2: Example of SenticNet semantic graph.

sentiment polarity score, which provide a quantified sentiment evaluation. Fig. 3.3(b) gives one example. *Princeton WordNet* [129] is a large lexical database in English. It organizes lexical elements in the form of synsets under four POS categories: Nouns, Verbs, Adjectives and Adverbs. The total 117k synsets were ordered in a hierarchical structure that made semantic inference possible. NTU MC (*NTU multi-lingual corpus*¹) [130] translates *Princeton WordNet* into over 30 languages. It is a multilingual corpus that contains 375k words [130]. It has 42k Chinese concepts in the corpus and are linked by corresponding English translations in WordNet. The concepts in Chinese NTU MC is manually aligned to English counterparts, which makes NTU MC an ideal bridge to map bilingual concepts. Furthermore, Chinese texts in NTU MC were translated by human experts into both single word and multi-word expressions, making it more than a lexical resource. *SentiWordNet* is a lexical resource which assigns a positive and a negative score to each synset in *WordNet*. We next introduce how the above resources were used to construct Csentinet.

3.3.2 Two Versions

Among all the resources we introduced above, only NTU MC is in the Chinese language. Therefore, it serves as the source of Chinese text. The downside is it does not have any information on sentiment. Thus, the general idea is to append affective information to NTU MC to make it a sentiment resource.

¹<http://compling.hss.ntu.edu.sg/ntumc/>

In terms of sentiment resources, we have SentiWordNet and SenticNet. Since they are independent of each other, we can use either of them to construct the sentiment resource. As such, we used SentiWordNet in the first version and then SenticNet in the second version.

In the first version, we extract sentiment information from SentiWordNet and append to NTU MC. Since SentiWordNet corresponds to each synset in WordNet and NTU MC is manually translated from WordNet, we combine Chinese text from NTU MC and sentiment score from SentiWordNet of each synset to form a Chinese sentiment resource.

In the second version, we extract sentiment information from SenticNet and append to NTU MC. In the beginning, we find all the single and multi-word expressions that can be directly found in both SenticNet and WordNet. This step is named direct mapping. In order to solve the non-matched concepts, we proposed enhanced mapping which jointly utilized POS tagging and extended Lesk algorithm (introduced later in Sec. 3.5.2.2) to solve the unmatched concepts. In the end, sentiment information from SenticNet of these matched concepts was combined with the Chinese translations from NTU MC to create the second version CsentNet.

3.4 First Version: SentiWordNet + NTU MC

The key idea of the first version is to transfer sentiment information from SentiWordNet to NTU MC. Specifically, we first map words from NTU MC to WordNet, followed by extracting sentiment score from SentiWordNet.

We start off by analyzing the structure of NTU MC. The knowledge base was organized in a lexical hierarchy. The root hierarchy is ‘LexicalResource’. Under the root node, there are two children branches: ‘Lexicon’ and ‘SenseAxes’. ‘Lexicon’ is the mother of 61k ‘LexicalEntry (ies)’. Each ‘LexicalEntry’ has a Chinese word, its POS, its Sense ID and synset. Because some Chinese words may have different meanings in English, these ‘LexicalEntries’ will have more than one pair of Sense ID and synset. Fig. 3.3(a) below shows an example. The key clue that links NTU MC to

<pre> <LexicalEntry id='w240003'> <Lemma writtenForm='售完' partOfSpeech='v'/> <Sense id='s240003_02208409-v' synset='cmn-10-02208409-v'/> </LexicalEntry> <LexicalEntry id='w223142'> <Lemma writtenForm='靠近' partOfSpeech='a'/> <Sense id='w223142_00444519-a' synset='cmn-10-00444519-a'/> <Sense id='w223142_00444984-a' synset='cmn-10-00444984-a'/> <Sense id='w223142_00447472-a' synset='cmn-10-00447472-a'/> </LexicalEntry> <LexicalEntry id='w229294'> <Lemma writtenForm='神经过敏+地' partOfSpeech='r'/> <Sense id='w229294_00409327-r' synset='cmn-10-00409327-r'/> </LexicalEntry> </pre>	<pre> <text>delicious meal</text> <semantics casserole/> <semantics meatloaf/> <semantics hot_dog_bun/> <semantics hamburger/> <semantics hot_dog/> <pleasantness> 0.028 </pleasantness> <attention> -0.0732 </attention> <sensitivity> 0 </sensitivity> <aptitude> 0 </aptitude> <polarity> 0.034 </polarity> </pre>
(a) NTU MC data	(b) SenticNet data

Figure 3.3: Example of used sentiment resources

WordNet is the synset ID. *synset=cmn-10-02208409-v* is a synset. The combination of *-02208409* and *-v* uniquely distinguish each synset (sense) both in the NTU MC and WordNet. Naturally, we re-organize the structure of this knowledge base by grouping all the words by synsets with unique synset ID. After processing, we have obtained 42k synsets and each synset has at least one Chinese word.

Then we process SentiWordNet by firstly combining *POS* and *ID* of each synset and writing them into the same format as NTU MC. Then we compute the sentiment polarity value of each synset. As each synset has a positive score and a negative score, we subtract the absolute value of negative score from the positive score and treat the result as the sentiment polarity score. The range of final score is between -1 and +1, where polarity represents sentiment polarity and absolute value means intensity.

In some cases, the calculation results can be 0. This is due to either the synset having neither positive nor negative sentiment or the synset having equal positive and negative scores. We eliminate these synsets since they express no strong sentiment. Even though this reduces the size of the resulting resource, the elimination of these synsets prevents introducing false information. The final version is in text format. Each line of the file has a synset (omitted in Figure) with its sentiment polarity score and the relevant Chinese words. Figure 3.1(b) shows some examples of the results.

3.5 Second Version: SenticNet + NTU MC

In the second version, we aim to transfer the sentiment information from SenticNet to NTU MC. Because NTU MC is directly related to WordNet and WordNet is much bigger than SenticNet, it makes more sense to map SenticNet to NTU MC than the other way around. Thus, the complete mapping contains these 3 steps: map NTU MC to WordNet, map SenticNet to WordNet, and extract the overlap between SenticNet's and NTU MC's mappings in WordNet.

As the first step of mapping NTU MC to WordNet was accomplished in the first version, we directly inherit from there. The last step of extracting the overlap is relatively straightforward. Thus, in this second version, we mainly focus on introducing the second step, namely how to map SenticNet to WordNet. Before that, we present an analysis of SenticNet below.

3.5.1 SenticNet and Preprocessing

As we can see from the Figure 3.3(b), the sentiment value of the multi-word concept is 0.034, which is a positive sentiment. The 5 semantics *casserole*, *meatloaf*, *hot_dog_bun*, *hamburger* and *hot_dog* all contribute to the concept of a delicious meal. We consider each of the semantics alone as sharing a similar sentiment value with the concept, but we give each concept a higher priority than its semantics. From SenticNet, we have extracted about 17,000 concepts. Before mapping, we need to preprocess SenticNet. We extract every concept, its 5 semantics and its sentiment score and then put them in a Python dictionary. The key of the dictionary is the concept, and the value is the corresponding semantics and sentiment score.

3.5.2 Mapping SenticNet to WordNet

After the preprocessing is done, we start step 2: mapping SenticNet to WordNet. Due to the diversity of SenticNet (single word, multi-word phrase, semantics), we have proposed two solutions to the problem: direct mapping and enhanced mapping. The direct mapping tries to map SenticNet to WordNet by word-to-word matching.

The enhanced mapping integrate direct mapping with keyword extraction based on POS and extended Lesk algorithm.

3.5.2.1 Direct Mapping

Since we have covered both SenticNet and WordNet in the Python dictionary, we can conduct mapping directly. With WordNet, we have obtained a Python dictionary which key is the word or phrase in WordNet and value is a list of synset ID.

For WordNet, a key-value pair is made up of concept followed by synset IDs, such as {activated : [cmn-10-01313587-a, cmn-10-01314680-a], ...}. In terms of SenticNet, a key-value pair is made up of concept followed by its semantics, for instance {bank : [coffer, bank_vault, finance, government_agreement, money], ...}. We match each key in SenticNet dictionary to each key from the WordNet dictionary. If a key was matched, the hypernyms of each synset ID in the value from the WordNet dictionary would be retrieved. Hypernyms are retrieved from WordNet itself. Synsets (hyponyms) are subordinates of their hypernyms. Then hypernyms of each synset ID will be matched with the words (both concept and semantics) in key-value pair from SenticNet.

If hypernyms from only one synset ID were matched, then this matched synset from WordNet shares the same meaning with the concept-semantics pair from SenticNet. Thus, the sentiment score of this concept from SenticNet will be given to this synset ID. If hypernyms from more than one synset ID were matched, we compute how many words are matched with hypernyms for each synset ID and choose the synset that has most matched words as final matched synset, which will be given the sentiment score from SenticNet. The hypernym of synset ID is considered as layer 1. Hypernyms of the previous hypernyms are considered as layer 2 so on and so forth. If nothing was matched through the whole concept-semantics list in layer 1, we proceed to layer 2. If nothing was matched after layer 3, a concept is scrapped. In the end, we accomplish mapping and obtain a dictionary whose key is the synset ID and value is the sentiment score.

The final dictionary has 12,042 key-value pairs, which means we have mapped 12,042 synsets from SenticNet to WordNet, a size about one-fourth of that of NTU MC. However, one issue that direct mapping failed to solve is the accuracy of matches. For example, referring to Figure 3.3(b), we have a concept *delicious meal* and a sentiment score of 0.034. We can see that the sentiment score strongly represents the word *delicious* rather than *meal*. However, due to its non-exact match to WordNet, we lose the sentiment score of *delicious meal*, as well as the word *delicious*. In order to figure out the above-mentioned issue, we have developed an enhanced mapping method on top of direct mapping.

3.5.2.2 Enhanced Mapping with POS Analysis and Extended Lesk Algorithm

As direct mapping has above problems, we develop POS analysis to tackle the exact match problem when the concept was not matched and combine extended Lesk algorithm to settle the problem of sense disambiguation when matching hypernyms failed.

Before POS analysis, we tokenize the phrases using the most popular third-party tool *Natural Language Toolkit*. Afterward, we annotate the tokens with the POS tag. It helps to extract the key meaning in terms of sentiment and to distinguish the usage of a word in its different senses. We again take the example from Figure 3.3(b). The concept *delicious meal* has a word *delicious* that is a POS adjective and a word *meal* which is a POS noun. The sentiment of this concept is expressed more by the adjective than the noun. By annotating the POS of each token, we have a better understanding of the sentiment of concept. Furthermore, it helps to distinguish different senses of a word. For example, *time flies* and *fruit flies* both have the word *flies*. In *time flies*, *flies* has a verb POS tag. In *fruit flies*, *flies* has a noun POS tag. The POS tag allows these two senses to be easily distinguished from the other. However, there are also cases there can be different senses of one word in within the same POS. In such situations, we need to apply the Lesk algorithm to solve the issue.

The Lesk algorithm is a word sense disambiguation algorithm developed by Michael Lesk in 1986 [131]. The algorithm is based on the idea that the sense of a word is

in accordance with the common topic of its neighborhood. A practical example used in word sense disambiguation may look like this. Given an ambiguous word, each of its sense definition in the dictionary is fetched and compared with its neighborhood text. The number of common words that appear in both the sense definition and neighborhood text is recorded. In the end, the sense that has the biggest number of common words is the sense of this ambiguous word. However, the ambiguous word may sometimes not have enough neighborhood text, so, people have developed ways to extend this algorithm. Timothy [132] explores different tokenization schemes and methods of definition extension. Inspired by their paper, we also developed a way of extension in our experiments. The extended algorithm can solve the ambiguous mapping problem in our direct mapping method.

All single words from SenticNet were easily matched to WordNet. The difficulty mainly falls in mapping multi-word phrases. We put a higher priority on the concepts than their semantics. The reason is that sentiment scores in SenticNet are specifically computed for the concepts. Its semantics carry the close-related meaning of the concept which share similar sentiment score. Strictly speaking, this is not ideal.

Therefore, like direct mapping, we decide to match each concept in SenticNet to WordNet first. If it was not matched, we annotate the concept (if it is multi-word phrase) tokens with POS tags before sorting them by POS tag priority. The POS tag priority, from top to bottom, is Verbs, Adjective, Adverb and Noun. This order of priority is based on the heuristics that top POS tags are more emotionally informative [133, 134]. The next step is to extend the contexts. We tokenize all 5 semantics of a concept and concatenate them with the concept token string to form a large token string. This string is considered as our extended context. At this point, we have prepared the necessary inputs for the Lesk algorithm.

The prioritized tokens with POS tags are considered as the ambiguous words while the large token string is the neighborhood text. We then treat the concept tokens one by one as ambiguous words, based on their POS priority, and apply these to the Lesk algorithm to compute the sense. Once the sense was matched to a sense in WordNet, the processing of this concept is finished and this sense and sentiment

Figure 3.4: Mapping Framework of SenticNet Version

score is stored. If it was not matched after iterating through the concept tokens, then one of its semantics is POS tagged and the earlier listed procedures repeated. This process will not stop until a match is found in WordNet or all 5 semantics have been iterated. Figure 3.4 summarizes the framework of our two-version method.

In the end, we obtained a dictionary with 18,781 key-value pairs of synsets mapped from SenticNet to WordNet. This gave us 6,739 more pairs than the direct mapping method.

3.5.3 Find and Extract the Overlap

From the previous section, we obtained a Python dictionary whose key-value pair is synset ID-sentiment score by mapping SenticNet to WordNet. In this section, we combine the dictionary with the NTU MC Python dictionary we got in the first version and find their overlap. In the end, 5,677 synsets were overlapped, which meant they had corresponding Chinese translations in NTU MC. Over 15,000 overlapped synsets with their sentiment score and Chinese translations were eventually written into a text file.

3.6 Evaluation

In this section, we conduct three evaluations of our mapping methods. For manual validation, we asked two native Chinese speakers to each evaluate 200 entries in our final text files for the two versions of Chinese sentiment resource. Particularly for

Table 3.1: Accuracy of SentiWordNet and SenticNet version (column 2 to 7) and accuracy of small value sentimnet synsets (last 3 columns)

Annotator	SentiWordNet version			SenticNet version			[-0.25, 0]	[0, 0.25]	Overall
	Positive	Negative	Overall	Positive	Negative	Overall			
1	48%	64%	56%	82%	80%	81%	75%	81%	78%
2	50%	58%	54%	78%	76%	77%	75%	83%	79%
Kappa measure	0.96	0.79	-	0.88	0.88	-	0.73	0.70	-

Table 3.2: Comparisons between CSenticNet and state-of-art sentiment lexicons

Sentiment resource	Chn2000			It168			Weibo		
	P	R	F1	P	R	F1	P	R	F1
NTUSD	50.08%	99.18%	66.55%	54.51%	97.66%	69.97%	51.17%	99.39%	67.56%
HowNet	53.29%	98.68%	69.21%	61.07%	96.79%	74.89%	50.76%	98.66%	67.03%
CSenticNet (SenticNet version)	54.85%	96.18%	69.86%	59.04%	94.19%	72.58%	55.90%	87.11%	68.10%

each of the two versions, 50 positive and 50 negative entries were randomly selected. Both experts were asked to label 200 entries from two versions as either positive or negative independently. We treat their manual labels as ground truth and compute the accuracies of our mapped sentiment resources. The results and inter-annotator agreement measures are in columns 2 to 7 of Table 3.1.

The results shown in the tables suggest that the SenticNet version outperforms the SentiWordNet version by almost 50 percent. This also validates our assumption that SenticNet is more reliable than SentiWordNet in terms of sentiment accuracy. As can be seen, the highest accuracy rate is over 80 percent. Moreover, there is still space to make improvements to this in the future.

In our mapping procedure, we assume synonyms and hypernyms share similar sentiment orientation with their root word. We believe this is true for the majority of words in the corpora. However, some words or expressions could have opposite sentiment orientation with their synonyms and hypernyms. As illustrated by the Hourglass model in [20], we know that words or expressions that have ambiguous sentiment orientation tend to have small absolute sentiment values. In order to validate our assumptions, we first inspect the sentiment value distribution of our SenticNet version sentiment resource and conduct manual validations.

Figure 3.5 presents the distribution of all synsets based on their sentiment values. An empty interval exists in the sentiment axis around zero value. This suggests no synsets have very small absolute sentiment values. It partially proves our initial

Figure 3.5: Distribution of sentiment values

assumptions. However, we notice the high intensity of synsets with small values just beyond the empty interval. The sentiment of these synsets could be wrongly mapped due to our synonym and hypernym assumptions. Thus, we randomly picked up 5 subsets of synsets from sentiment value ranges $(-0.25, 0]$ and $(0, 0.25]$, respectively. Each subset contains 20 synsets. Then we asked the two native Chinese speakers to label sentiment orientation of the 200 chosen synsets and treat their labels as ground truth. Results are shown in last 3 columns of Table 3.1. Accuracies within the chosen intervals keep abreast with that of the whole axis. According to the second expert, the intervals even outperform the whole axis in sentiment orientation prediction. Furthermore, we find that kappa measures of these intervals are less confident than that of the whole axis (columns 3 to 7 in Table 3.1). These results further support our initial assumptions and guaranteed the accuracy of our proposed sentiment resources.

Last but not least, shown in Table 3.2, we conduct sentiment analysis experiments to compare our CSenticNet (SenticNet version) with state-of-art baselines, HowNet and NTUSD. Three datasets we used are Chn sentiment corpus 2000 (Chn2000²), It168³ and Weibo dataset from NLP&CC⁴. The first dataset contains reviews from hotel customers. We preprocess this dataset by manually selecting only one sentence

²http://searchforum.org.cn/tansongbo/corpus/ChnSentiCorp_htl_ba.2000.rar

³<http://product.it168.com>

⁴NLP&CC is an annual conference of Chinese information technology professional committee organized by Chinese computer Federation (CCF). More details are available at <http://tcci.ccf.org.cn/conference/2013/index.html>

which has clear sentiment orientation from each review. The second dataset contains 886 reviews of digital product downloaded and manually labeled from a Chinese digital product website. The third dataset was micro-blogs originally used for opinion mining. We manually selected and labeled 1900 positive and 1900 negative sentences, respectively. We use a simple rule-based keyword matching classifier for testing. For a testing sentence, the classifier matches each of its words in sentiment lexicon and sums up the sentiment polarity of matched words in the sentence. For the baselines, positive words have +1 polarity and negative words have -1 polarity. If the final sum is above zero, then the sentence is positive and vice versa.

We see that CSenticNet outperforms the other two baselines in Chn2000 and Weibo datasets, at it has both higher precision and F1 score. However, it narrowly falls behind HowNet in the It168 dataset. We believe this is because of the highly domain-biased dataset. It168 reviews are mostly in digital fields, but CSenticNet is not tuned for that domain. Thus, it was not supposed to defeat the other two baselines but even though it still performs better than NTUSD. We also find that the recall of CSenticNet is not high, and this gives us a chance to further enlarge the resource by using new versions of SenticNet in the future.

3.7 Summary

In this chapter, we introduced an approach to building a concept-level Chinese sentiment resource. The approach contains mapping algorithms that make use of both Chinese and English multi-lingual corpus. It also tackles sense ambiguity problem and non-exact match issues. Depending on the two types of English sentiment resource used (SentiWordNet and SenticNet), we provide two versions of Chinese sentiment resource. The resource is organized in a semantic graph structure which makes word inference possible. Furthermore, each concept in the resource was described by a fine-grained sentiment polarity score. The SenticNet version achieves the state-of-the-art performance in the experimental evaluations.

Chapter 4

Radical-Based Hierarchical Embeddings

4.1 Introduction

For every NLP task, text representation is always the first step. In English, words are segmented by spaces and they are naturally taken as basic morphemes in text representation. Then, word embeddings were born based on distributed hypothesis.

Unlike English, whose fundamental morpheme is a combination of characters, such as prefixes, words etc., the fundamental morpheme of Chinese is radical, which is a (graphical) component of Chinese characters. Each Chinese character can contain up to five radicals. The radicals within character have various relative positions. For instance, it could be left-right (‘蛤 (toad)’, ‘秒 (second)’), up-down (‘岗 (hill)’, ‘孬 (not good)’), inside-out (‘国 (country)’, ‘问 (ask)’ etc. Radicals have two main functions: pronunciation and meaning. As the aim of this work is sentiment prediction, we are more interested in the latter function. For example, the radical ‘疒’ carries the meaning of disease. Any Chinese character containing this radical is related with disease and, hence, tends to express negative sentiment, such as ‘病 (illness)’, ‘疯 (madness)’, ‘瘫 (paralyzed)’, etc. In order to utilize this semantic and sentiment information among radicals, we decide to map radicals to embeddings (numeric representation at lower dimension).

The reason why we chose embeddings rather than classic textual feature like n-gram, POS, etc. is because that the embedding method is based on the distributed

hypothesis, which greatly explores the semantics and relies on token sequences. Correspondingly, radicals alone may not carry enough semantic and sentiment information. It is only when they are placed in a certain order that their connection with sentiment begins to reveal [3].

To the best of our knowledge, no sentiment-specific radical embeddings have ever been proposed before this work. We firstly train a pure radical embedding named R_{semantic} , hoping to capture semantics between radicals. Then, we train a sentiment-specific radical embedding and integrate it with the R_{semantic} to form a radical embedding termed R_{sentic} , which encodes both semantic and sentiment information [37, 103]. With the above, we integrate the two obtained radical embeddings with Chinese character embedding to form the radical-based hierarchical embedding, termed H_{semantic} and H_{sentic} , respectively.

The rest of the chapter is organized as follows: Section 4.2 presents a detailed analysis of Chinese characters and radicals via decomposition; Section 4.3 introduces our hierarchical embedding models; Section 4.4 demonstrates experimental evaluations of the proposed methods; finally, Section 4.5 concludes the chapter.

4.2 Background

Chinese written language dates back to 1200-1050 BC from the Shang dynasty. It originates from an Oracle bone script, which was iconic symbols engraved on ‘dragon bones’. From this time on was the first stage of Chinese written language development, Chinese written language was completely pictogram. However, different areas within China maintained different set of writing systems.

The second stage started from the unification in Qin dynasty. Seal script, which was an abstraction of the pictogram, became dominating over the empire from then on. Another apparent characteristic during this time was new Chinese characters were invented by combinations of existing and evolved characters. Under the mixed influence of foreign culture, development of science and technology and the evolution of social life, a great deal of Chinese characters were created during this time.

One feature of these characters is that they are no longer pictograms, but they are decomposable. Each of the decomposed elements (or radicals) carries a certain function. For instance, ‘声旁 (phoneme)’ labels the pronunciation of this character and ‘形旁 (morpheme)’ symbolizes the meaning of this character. Further details will be discussed in the following section.

The third stage occurred in the middle of the last century when the central government started advocating simplified Chinese. The old characters were simplified by reducing certain strokes within the character. The simplified Chinese characters dominate over mainland China ever since. Only Hong Kong, Taiwan and Macau retain the traditional Chinese characters.

4.2.1 Chinese Radicals

Due to the second stage of the above discussion, all modern Chinese character can be decomposed to radicals. Radicals are graphical components of characters. Some of the radicals in the character act like phonemes. For example, the radical ‘丙’ appears in the right half of character ‘柄 (handle)’ and symbolizes the pronunciation of this character. People even sometimes can correctly predict the pronunciation of a Chinese character which he or she does not know by recognizing certain radicals inside.

Some other radicals in the character act like morphemes that carry the semantic meaning of the character. For example, ‘木 (wood)’ itself is both a character and a radical. It means wood. A character ‘林 (jungle)’ which is made up of two ‘木’ means jungle. A character ‘森 (forest)’ which is made up of three ‘木’ means forest. In another example, radical ‘父’ is a formal form of word ‘father’. It appears on top of character ‘爸’ and this character means father exactly, but less formal, like ‘dad’ in English.

Moreover, the meaning of a character could be concluded from a integration of its radicals. A good example given by [43] is character ‘朝’. This character is made up of ‘十’, ‘日’, ‘十’ and ‘月’ four radicals. These four radicals are evolved from pictograms. ‘十’ stands for grass. ‘日’ stands for the sun. ‘月’ stands for the

moon. The integration of these four means the sun replaces the moon on the grass land, which is essentially the word ‘morning’. Not surprisingly, the meaning of this character ‘朝’ is indeed morning. This could continue. If the radical ‘氵’ which means water was attached to the left of character ‘朝’, then it is another character ‘潮’. Literally, this character means the water coming up in the morning. In fact, this ‘潮’ means tide, which matches its literal meaning.

To conclude, radicals entail more information than characters alone. Character-level research can only study the semantics expressed by characters. However, deeper semantic information and clues could be found at radical-level analysis [135]. This motivates us to apply deep learning techniques to extract this information. Prior to that, as we discussed in the related works, most works are in English language. Since English is very different from Chinese in many aspects, especially in decomposition, we have shown a comparison in Table 1.1.

As we could see from Table 1.1, character level is the minimum composition level in English. However, the equivalent level in Chinese is one level down than character, which is the radical level. Unlike English, semantics are hidden within each character in Chinese. Secondly, the Chinese word can be made up of single character or multi-character. Moreover, there is no space between words in a Chinese sentence. All the above observations indicate that normal English word embedding cannot be directly applied to Chinese. Extra processing like word segmentation, which will introduce errors, need to be conducted first.

Furthermore, if a new word or even a new character is out-of-vocabulary (OOV), normal word-level or character-level have no reasonable solution except giving a random vector. In order to address the above issues and also to extract the semantics within Chinese characters, a radical-based hierarchical Chinese embedding method is proposed in this chapter.

4.3 Hierarchical Chinese Embedding

In this section, we first introduce the deep neural network used in training our hierarchical embeddings. Then, we discuss our radical embedding. Finally, we present the

hierarchical embedding model.

4.3.1 Skip-Gram Model

We employ the Skip-gram neural embedding model proposed by [115] together with the negative sampling optimization technique in [121]. In this section, we briefly summarize the training objective and the model. Skip-gram model can be understood as a one-word context version CBOW model [115] working over C panels, where C is the number of context words of target word. Opposite to CBOW model, the target word is at the input layer whereas context words are at the output layer. By generating the most probable context words, the weight matrix can be trained and embedding vectors can be extracted.

Specifically, it is a one hidden layer neural network [136]. For each input word, it was denoted with an input vector V_{wi} . The hidden layer is defined as:

$$h = V_{wi}^T$$

where h is the hidden layer, wi is the i th row of input-hidden weight matrix W . At the output layer, C multinomial distributions were output, given each of the output is computed with the hidden-output matrix as:

$$p(w_{c,j} = w_{O,c} | w_i) = y_{c,j} = \frac{\exp(u_{c,j})}{\sum_{j'=1}^V \exp(u_{j'})}$$

where $w_{c,j}$ is the j th word on the c th panel of the output layer; $w_{O,c}$ is the c th word in the output context words; w_i is the input word vector; $y_{c,j}$ is the output of the j th unit on the c th panel of the output layer; $u_{c,j}$ is the net input of the j th unit on the c th panel of the output layer. Furthermore the objective function is to maximize the formula below:

$$\sum_{(w,c) \in \mathbb{D}} \sum_{w_j \in c} \log P(w | w_j)$$

where w_j is the j th word in contexts c , given the target word w .

4.3.2 Radical-Based Embedding

Traditional radical researches like [42] only take out one radical from each character to improve the Chinese character embedding. Moreover, to the best of our knowledge, no sentiment-specific Chinese radical embedding has ever been proposed yet. Thus, we propose the following two radical embeddings for Chinese sentiment analysis.

Inspired by the facts that Chinese characters can be decomposed to radicals and these radicals carry semantic meanings, we directly break characters into radicals and concatenate them in the order from left to right. We treat the radicals as the fundamental units in texts. Specifically, for any sentence, we decompose each character into its radicals and concatenate these radicals from different characters as a new radical string. Then we did the above preprocessing to all sentences in the corpus. Finally, a radical-level embedding model is built on this radical corpus using skip-gram model. We call this type of radical embedding as semantic radical embedding (R_{semantic}) because the major information extracted from this type of corpus is semantic between radicals. In order to extract the sentiment information between radicals, we developed the second type of radical embedding which is sentic radical embedding (R_{sentic}).

After studying the radicals, we have found that radicals themselves do not convey much sentiment information. What carries the sentiment information is the sequence or combination of different radicals. Thus, we take advantages of existing sentiment lexicons as our resource to study the sequence. Like we did before, we collect all the sentiment words from two different popular Chinese sentiment lexicons, HowNet [39] and NTUSD [40] and break them into radicals. Then we employ the skip-gram model to learn the sentiment related radical embedding ($R_{\text{sentiment}}$).

Since we want the radical embedding have both semantic information and sentiment information, we therefore conduct a fusion process of the previous two embeddings. The fusion formula is given as:

$$R_{\text{sentic}} = (1 - \omega) \cdot R_{\text{semantic}} + \omega \cdot R_{\text{sentiment}}$$

where R_{sentic} is the resulting radical embedding that integrates both semantic and sentiment information; R_{semantic} is the semantic embedding and $R_{\text{sentiment}}$ is the sentiment

Figure 4.1: Performance on four datasets at different fusion parameter

embedding; w is the weight of the fusion. If w equals to 0, then the R_{sentic} is pure semantic embedding. If w equals to 1, then the R_{sentic} is pure sentiment embedding.

In order to find the best fusion parameter, we conduct tests on separated development subsets of four real Chinese sentiment datasets, namely: Chn2000, It168, Chinese Treebank [84] and Weibo dataset (details in next section). We train a convolutional neural network (CNN) to classify the sentiment polarity of sentences in the datasets. The features we use are the sentic radical embedding, but we apply the features at different fusion parameter value. The classification accuracies of different fusion values on four datasets are shown in Fig. 4.1. As the heuristics from Fig. 4.1 suggest, we take the fusion parameter of value 0.7 which performs best.

4.3.3 Hierarchical Embedding

Hierarchical embedding is based on the assumption that different levels of embeddings will capture different level of semantics. According to the hierarchy of Chinese in Table 1.1, we have already explored the semantics as well as sentiment at the radical level. The next higher level is the character level, followed by the word level (multi-character word). However, we only select character-level embedding ($C_{semantic}$) to be integrated into our hierarchical model because characters are naturally segmented by Unicode (no pre-processing or segmentation needed). Although existing Chinese word segmenter could achieve certain accuracy, it can still introduce segmentation errors that affect the performance of word embedding. In the hierarchical model, we also use the skip-gram model to train independent Chinese character embeddings.

Then we fuse the character embeddings with either the semantic radical embedding (R_{semantic}) and the sentic radical embedding (R_{sentic}) to form two types of hierarchical embeddings: H_{semantic} and H_{sentic} , respectively. The fusion formula is the same with that in radical embeddings, except that with a different fusion parameter value of 0.5 based on our development tests. A graphical illustration depicted in Fig. 4.2.

Figure 4.2: Framework of hierarchical embedding model

4.4 Experimental Evaluation

We evaluate our proposed method on Chinese sentence-level sentiment classification task in this section. Firstly, we introduce the datasets used for evaluations. Then, we demonstrate the experimental settings. Lastly, we present the experimental results and provide an interpretation for them.

4.4.1 Dataset

There are four sentence-level Chinese sentiment datasets used in our experiments. The first is Weibo dataset (Weibo) which is a collection of Chinese microblogs from NLP&CC, with about 2000 blogs for either positive or negative category. The second dataset is a Chinese Tree Bank (CTB) introduced by [84]. For each sentiment category, we have obtained over 19000 sentences after mapping their sentiment values to polarity. The third dataset Chn2000, contains about 1339 hotel reviews from customers¹. The last dataset IT168, have around 1000 digital product reviews². All

¹<http://searchforum.org.cn/tansongbo/corpus>

²<http://product.it168.com>

the above datasets are labeled as positive or negative at the sentence level. In order to prevent overfitting, we conduct 5-fold cross validations on all our experiments.

4.4.2 Experimental Setting

As embedding vectors are usually used as features in classification tasks, we compare our proposed embeddings with three baseline features: character-bigram, word embeddings and character embeddings. In choosing the classification model, we take advantage of state-of-the-art machine learning toolbox scikit-learn [94].

Four classic machine learning classifiers were applied in our experiments: LinearSVC (LSVC), logistic regression (LR), Naïve Bayes classifier with a Gaussian kernel (NB) and multi-layer perceptron (MLP) classifier. In evaluating the embedding features on these classic machine learning classifiers, an average embedding vector is computed to represent each sentence, given a certain granularity of the sentence cells. For instance, if a sentence is broken into a string of radicals, then the radical embedding vector of this sentence is the arithmetic mean (average) of its component radical embeddings. Furthermore, we apply CNN in the same way proposed in [25], except that we reduce the embedding vector dimension to 128.

4.4.3 Results and Discussion

Table 4.1 compared bigram feature with semantic radical embedding, sentic radical embedding, semantic hierarchical embedding and sentic hierarchical embedding using five classification models on four different datasets. Similarly, Table 4.2 also compared the proposed embedding features with word2vec and character2vec features. In all of the four datasets, our proposed features working with CNN classifier achieved the best performance. In Weibo dataset, sentic hierarchical embedding performed slightly better than character2vec, with less than 1% improvement. However in CTB and Chn2000 datasets, semantic hierarchical beat three baseline features by 2~6%. In the IT168 dataset, the sentic hierarchical embedding was second to bigram feature in the MLP model.

		Bigram(%)			Rsemantic(%)			Rsentent(%)			Hsemantic(%)			Hsentent(%)		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Weibo	LSVC	71.65	71.60	71.58	66.77	66.64	66.57	67.46	67.35	67.30	71.02	70.94	70.91	72.74	72.66	72.63
	LR	74.38	74.32	74.30	65.51	65.28	65.15	65.29	65.12	65.02	70.39	70.31	70.27	72.47	72.37	72.33
	NB	63.84	63.01	62.15	57.60	55.74	52.90	58.67	56.73	54.21	59.16	55.97	51.74	60.42	57.63	54.58
	MLP	72.54	72.50	72.48	67.02	66.93	66.89	67.31	67.25	67.22	70.53	70.49	70.47	73.03	73.00	72.99
	CNN	-	-	-	75.27	73.71	73.19	75.44	75.41	75.38	73.88	72.91	72.55	75.82	75.60	75.58
CTB	LSVC	76.45	76.32	76.29	67.22	67.19	67.17	66.34	66.28	66.25	68.57	68.55	68.54	69.15	69.11	69.10
	LR	78.12	77.99	77.97	65.29	65.25	65.22	64.91	64.85	64.81	68.25	68.22	68.21	69.24	69.20	69.19
	NB	66.60	62.80	60.46	60.99	60.43	59.86	60.41	59.59	58.64	61.24	60.04	58.90	63.52	62.50	61.74
	MLP	76.13	76.01	75.98	67.71	67.68	67.66	66.92	66.79	66.72	70.98	70.96	70.95	70.01	69.78	69.69
	CNN	-	-	-	77.68	77.67	77.65	79.59	79.42	79.42	80.77	80.77	80.76	80.79	80.69	80.65
Chn2000	LSVC	82.43	82.21	82.22	70.64	67.32	67.12	66.00	61.26	59.70	73.73	72.74	72.87	74.57	73.61	73.71
	LR	83.22	82.68	82.76	69.99	55.50	48.04	68.50	51.34	39.51	70.62	67.65	67.50	72.38	68.24	67.86
	NB	67.06	66.68	65.68	67.23	66.93	66.54	63.34	63.36	62.95	64.93	64.44	64.33	67.92	67.75	67.59
	MLP	80.71	80.42	80.47	69.00	68.57	68.59	67.47	66.85	66.83	74.00	73.63	73.62	73.23	73.06	73.05
	CNN	-	-	-	79.96	81.83	80.14	82.01	83.50	82.47	87.45	86.71	87.02	86.06	87.07	86.12
IT168	LSVC	81.95	82.06	81.93	72.53	70.23	70.18	72.72	69.85	69.71	79.55	79.00	79.11	80.77	80.30	80.44
	LR	83.86	83.72	83.74	71.40	60.82	57.32	73.58	56.10	48.47	77.58	75.46	75.71	79.46	76.80	77.09
	NB	63.84	63.01	62.15	64.73	63.62	63.45	63.50	62.62	62.46	67.75	66.21	66.12	71.90	70.09	70.16
	MLP	83.35	83.35	83.29	71.83	71.04	71.08	73.86	72.71	72.80	78.10	77.70	77.68	79.48	79.31	79.27
	CNN	-	-	-	84.38	84.33	84.33	83.95	83.87	83.83	85.39	84.50	84.07	83.75	83.43	83.15

Table 4.1: Comparison with traditional feature on four datasets

		W2V(%)			C2V(%)			Rsemantic(%)			Rsentent(%)			Hsemantic(%)			Hsentent(%)		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Weibo	LSVC	74.46	74.38	74.35	74.12	73.98	73.94	66.77	66.64	66.57	67.46	67.35	67.30	71.02	70.94	70.91	72.74	72.66	72.63
	LR	73.91	73.72	73.66	73.60	73.43	73.37	65.51	65.28	65.15	65.29	65.12	65.02	70.39	70.31	70.27	72.47	72.37	72.33
	NB	60.63	57.97	55.15	61.04	58.08	55.02	57.60	55.74	52.90	58.67	56.73	54.21	59.16	55.97	51.74	60.42	57.63	54.58
	MLP	73.68	73.58	73.55	74.49	74.43	74.41	67.02	66.93	66.89	67.31	67.25	67.22	70.53	70.49	70.47	73.03	73.00	72.99
	CNN	72.57	72.55	72.52	75.15	75.11	75.11	75.27	73.71	73.19	75.44	75.41	75.38	73.88	72.91	72.55	75.82	75.60	75.58
CTB	LSVC	71.15	71.12	71.11	68.92	68.90	68.90	67.22	67.19	67.17	66.34	66.28	66.25	68.57	68.55	68.54	69.15	69.11	69.10
	LR	70.87	70.84	70.83	68.50	68.48	68.47	65.29	65.25	65.22	64.91	64.85	64.81	68.25	68.22	68.21	69.24	69.20	69.19
	NB	67.56	67.51	67.49	63.49	62.61	61.96	60.99	60.43	59.86	60.41	59.59	58.64	61.24	60.04	58.90	63.52	62.50	61.74
	MLP	71.17	71.16	71.15	69.78	69.54	69.44	67.71	67.68	67.66	66.92	66.79	66.72	70.98	70.96	70.95	70.01	69.78	69.69
	CNN	78.56	78.56	78.56	78.56	77.93	77.75	77.68	77.67	77.65	79.59	79.42	79.42	80.77	80.77	80.76	80.79	80.69	80.65
Chn2000	LSVC	81.05	79.77	80.05	72.04	70.73	70.85	70.64	67.32	67.12	66.00	61.26	59.70	73.73	72.74	72.87	74.57	73.61	73.71
	LR	78.87	74.74	74.96	70.32	64.29	63.00	69.99	55.50	48.04	68.50	51.34	39.51	70.62	67.65	67.50	72.38	68.24	67.86
	NB	72.25	71.25	71.34	69.62	69.55	69.44	67.23	66.93	66.54	63.34	63.36	62.95	64.93	64.44	64.33	67.92	67.75	67.59
	MLP	79.53	79.18	79.24	70.84	70.65	70.67	69.00	68.57	68.59	67.47	66.85	66.83	74.00	73.63	73.62	73.23	73.06	73.05
	CNN	82.50	82.50	82.50	85.77	86.21	85.95	79.96	81.83	80.14	82.01	83.50	82.47	87.45	86.71	87.02	86.06	87.07	86.12
IT168	LSVC	82.43	81.15	81.46	78.68	77.80	78.00	72.53	70.23	70.18	72.72	69.85	69.71	79.55	79.00	79.11	80.77	80.30	80.44
	LR	82.11	77.73	78.11	77.79	72.69	72.67	71.40	60.82	57.32	73.58	56.10	48.47	77.58	75.46	75.71	79.46	76.80	77.09
	NB	60.63	57.97	55.15	71.12	69.78	69.89	64.73	63.62	63.45	63.50	62.62	62.46	67.75	66.21	66.12	71.90	70.09	70.16
	MLP	79.93	79.65	79.70	78.52	78.36	78.35	71.83	71.04	71.08	73.86	72.71	72.80	78.10	77.70	77.68	79.48	79.31	79.27
	CNN	82.23	81.50	81.40	82.69	82.63	82.65	84.38	84.33	84.33	83.95	83.87	83.83	85.39	84.50	84.07	83.75	83.43	83.15

Table 4.2: Comparison with embedding features on four datasets

This result was not surprising because bigram feature can be understood as a sliding window with a size of 2. Using the multi-layer perceptron classifier, the performance could be parallel to that of a CNN classifier. Even though, the other three proposed features working with CNN classifier beat all baseline features with any classifier. In addition to the above observations, we obtained the following analysis.

Firstly, deep learning classifiers worked best on embedding features. The performance of all embedding features reduced sharply when applying on classic classifiers. Nevertheless, even if the performance of our proposed features in classic machine learning classifiers dropped greatly compared with CNN, they still paralleled or beat other baseline features. Moreover, the performance of the proposed features was never

fine-tuned. Better performance can be expected after future fine-tuning.

Secondly, the proposed embedding features do unveil certain information that can promote sentence-level sentiment analysis. Although we were not certain where exactly the extra information located, because the performance of our four proposed embedding features were not robust (no single one feature achieved the best performance over all four datasets), we proved radical-level embedding contribute to Chinese sentiment analysis.

4.5 Summary

In this chapter, we proposed Chinese radical-based hierarchical embeddings particularly designed for sentiment analysis. Four types of radical-based embeddings were introduced: radical semantic embedding, radical sentic embedding, hierarchical semantic embedding and hierarchical sentic embedding. By conducting sentence-level sentiment classification experiments on four Chinese datasets, we proved the proposed embeddings outperform state-of-the-art textual and embedding features. Most importantly, our study presents the first piece of evidence that Chinese radical-level and hierarchical-level embeddings can improve the Chinese sentiment analysis.

Chapter 5

Multi-grained Aspect Target Sequence Modeling

5.1 Introduction

Aspect-based sentiment analysis (ABSA) proposes a finer-grained polarity detection that extracts aspects first and then classifies them as either positive or negative. For example, in the sentence “*The size of the room was smaller than our expectation but the view from the room would not make you disappointed.*”, sentiments expressed towards “*room size*” and “*room view*” are negative and positive, respectively. Those two terms are called aspect terms and ABSA associates a polarity to each aspect term. Another similar yet different sub-task of ABSA is sentiment analysis towards aspect category [33]. For example, both “*room size*” and “*room view*” in the previous example belong to “ROOM_FACILITY”. Other aspect categories in this domain are like “PRICE”, “SERVICE” and so on.

In this chapter, we focus on aspect term sentiment classification, which is a finer-grained study compared to the work of Wang et al [33]. We define aspect term as aspect target. If an aspect term contains multiple words, we call this type of aspect term as the aspect target sequence. In aspect target sentiment classification, Tang et al. [45] used target dependent an LSTM network. In particular, they use a Bi-LSTM model to encode the sequential information in TC-LSTM. They later appended each word with target embedding to reinforce the extraction of correlation between target and context words in the sentence. In [30], they designed a pure attention-based memory network to explicitly learn the correlation between context words and aspect

target. Nevertheless, they simply used average aspect word embedding to represent aspect term, which failed to consider the aspect target sequence information. Wang et al. [33] employed an attention mechanism upon the sequential output from an LSTM layer. Their work treated the sentence sequential information as equal importance to aspect target sequential information.

All the previous work modeled ABSA as a sentence-level sentiment classification problem that treated aspect target/term as a hint. Such design will result in a dilemma when there appear two aspect targets with opposite sentiment polarities in the same sentence. All state-of-the-art works only focused on one aspect target at a time. They cannot process two aspect targets at the same time, due to the assumption that the sentiment of a sentence is equivalent to the sentiment of aspect target (term). Moreover, little attention is paid to aspect target itself, especially when aspect target is a sequence of words, namely multi-word aspect. Almost all literature took the average word embeddings to represent aspect target sequence, which ignored aspect target sequential information. In the English language, their models work well in situations where aspect target has a single word, but not in multiple words. In other cases, although they employed a sentence-level sequence encoder, the importance of aspect target sequence is treated with no emphasis compared with non-aspect word sequence. To this end, we propose two versions of an aspect target sequence model (ATSM), namely: ATSM-S, where -S stands for single granularity, and ATSM-F, where -F stands for fusion.

ATSM-S explicitly addresses to the multi-word aspect target case. The model includes two crucial modules: adaptive embedding learning and aspect target sequence learning. The first module aims at appending sentence context meaning to general word embeddings for each of the aspect target words. Thus, an accurate vector representation which encoded sentence context will be obtained for aspect target words. Specifically, we extract sentence context with an LSTM encoder. Each aspect target word was attended by the encoded context to form an adaptive word embedding. The second module links each adaptive aspect word embedding with a sequence learning. In the experimental comparison, our ATSM-S outperforms the state of the art on an

English multi-word aspect subset filtered from SemEval 2014 and four Chinese review datasets.

Even though ATSM-S only solves part of the problems (multi-word aspect scenario) in English ABSA, it becomes comprehensive in addressing Chinese ABSA if considering the multi-granularity representation of Chinese text. Chinese is a pictogram language whose text originates from images. Chinese text originates from simple symbols. The symbols gradually evolved to fixed types (named radicals). Through a geometric composition, those fixed types build up characters. Then a concatenation of characters creates the word. Unlike English, each Chinese sub-word granular representation still encodes semantics, shown in Table 1.1. Whereas in English, only partial character N-grams encode semantics. This motivates us to explore each granularity of Chinese text in ABSA. In addition, the surface form of Chinese text is at the character level. This guaranteed that even the smallest aspect target, such as a single Chinese character, can be broken down into a sequence of aspect targets at the radical level. Thus, we proposed ATSM-F as an upgraded version of ATSM-S. Specifically, ATSM-S was conducted at each Chinese granularity and ATSM-F fused their results together. In the design of fusion, we tested both early fusion (hierarchical structure) and late fusion (flat structure). Finally, ATSM-F with late fusion prevails all other methods on three out of four Chinese review datasets. To round up, we made the following contributions:

- We view aspect-level sentiment analysis from a new perspective, in which aspect target sequence dominates the final result. Whereas in recent literature using deep learning, sentence-level classification is the popular solution [33, 45, 30].
- We propose the adaptive embedding learning to append sentence context to aspect targets. Followed by an explicit modeling of the aspect target sequence. Results on English multi-word aspect subset from SemEval2014 and four Chinese review datasets validate the superiority of our model.
- We leverage the multi-grained representation nature of Chinese text and improve the final performance further, which suggests a broader application scenario.

5.2 Background

In ABSA, there are three research directions. The first direction is aspect term extraction, such as [137, 8]. The second direction aims at categorizing the given aspect term to different categories [138, 139]. Wang et al. [33] employed an attention mechanism upon the sequential output from an LSTM layer. Their work aims at predicting sentiment polarity of the category, such as “FOOD” and “PRICE”, rather than any particular aspect terms.

The third branch works on aspect term sentiment classification. The aspect term is marked in a sentence and the goal is to determine the sentiment polarity towards the aspect term. Early works used dictionary-based methods [140, 141]. Recent works employed machine learning-based feature engineering and classification [142, 143].

Most state-of-the-art works used an LSTM network [144] and attention mechanism as the basic modules in their methods [30, 45]. Tang et al. used target dependent Bi-LSTM model to encode the sequential information in TC-LSTM. They later appended each word with target embedding to reinforce the extraction of correlation between target and context words in the sentence. In MemNet [30], they designed a pure attention-based memory network to explicitly learn the correlation between context words and aspect words.

Previous works on aspect-term sentiment analysis suffered from two main drawbacks. Firstly, ABSA is modeled as a sentence-level sentiment classification problem that treated aspect target/term as a hint. Such design will result in a dilemma when there appear two aspect targets with opposite sentiment polarities. All state-of-the-art works only focused on one aspect target at a time. They cannot process two aspect targets at the same time, due to the assumption that the sentiment of a sentence is equivalent to the sentiment of aspect target/term. Secondly, little attention is paid to aspect target itself, namely aspect target sequence information. In this chapter, we aim to address these two drawbacks.

5.3 Method Overview

In this section, we first define our task and then present an overview of the proposed method.

5.3.1 Aspect Target Sequence

Aspect is a concept that contains various interpretations, such as aspect target/term, aspect word, aspect category, aspect sentiment etc. For instance, a sentence “这菜味道不错。” (This cuisine has a good flavor.)” has an aspect target/term “味道 (flavor)”. The aspect target contains only one aspect word, which is “味道 (flavor)”. The aspect target belongs to an aspect category of “FOOD”. Other aspect categories in the domain of restaurant are like “PRICE”, “SERVICE” and so on. The sentiment of the aspect target “口味 (flavor)” is positive in the sentence.

However, in the context of this chapter, we define aspect as aspect target sequence. As Chinese text can be decomposed to three granularities, a single unit of higher level representation can be decomposed to a sequence of units of lower level representation. For instance, the single-word aspect target “味道 (flavor)” in the previous example can be decomposed to a sequence of Chinese characters: “味” and “道”. Moreover, the characters can be further decomposed to a sequence of Chinese radicals: “口”, “未”, “辶” and “首”. As [119, 44, 145] suggested, various granularities contain exclusive semantics. In the above example, “味道” at word level simply means ‘flavor’. “味” and “道” at character level mean ‘thinking of the flavor’. “口”, “未”, “辶” and “首” at radical level mean ‘to taste the unknown and brainstorm the flavor’. It is apparent to see from the example that sub-component semantics provide complementary explanations to the word and, hence, enrich its meaning. We reconstruct an aspect target as three sequences at three granularities. Methods were developed to work on these sequences in order to determine the sentiment polarity of the aspect target.

5.3.2 Task Definition

A sentence s of n units (the unit could be radical, character or word) in the format $s = \{u_1, u_2, \dots, u_j, u_{j+1}, \dots, u_{j+L}, \dots, u_{n-1}, u_n\}$ is marked out with aspect target (comprising

multiple units) $\{u_j, u_{j+1}, \dots, u_{j+L}\}$. The u_{j+L} stands for the $(j + L)th$ unit in the sentence and the Lth unit in the aspect target. L indicates that the aspect target contains L consecutive units. The goal is to predict the sentiment polarity of the aspect target.

Figure 5.1: ATSM-F late fusion framework. RNN-1,-2,-3 are at word, character and radical level, respectively. Green RNNs are for adaptive embedding learning. Grey RNNs are sequence learning of aspect target. Aspect target is highlighted in red.

5.3.3 Overview of the Algorithm

In all previous works of ABSA [30, 33, 45], if an aspect target contains multiple words, they treated the multi-word aspect as one unified target by averaging their word embeddings. This is disadvantageous in two ways. Firstly, word embeddings of aspect target that were trained from general corpus might mislead the meaning of aspect target in the sentence. Secondly, sequential information within the aspect target is lost. For instance, a sentence “The **red apple** released in California last year was a disappointment.” contains an aspect target “**red apple**”. Based on the occurrence of “released in California”, we can understand that “**red apple**” stands for iPhone. If general word embeddings of “**red**” and “**apple**” were used in the task, it will deviate from the symbolic meaning of iPhone to fruit apple. To make it worse,

by averaging the word embeddings of “*red*” and “*apple*”, sequential information is lost and the averaged word embedding will result in a new/irrelevant meaning in the word vector space.

In order to address the above two issues, we propose a three-step model. The first step is adaptive embedding learning, which essentially aims at learning intra-sentence context for each unit in the aspect target sequence. It was designed to embed intra-sentence contexts to the general embeddings of units in the aspect target sequence, which will resolve the first issue aforementioned. The second step is simply a sequence learning process of aspect target, which has never been addressed before. Last but not the least, Chinese text has three granularities of representation (radical, character and word) so that we apply the first two steps at each of the three granularities and glue them together with fusion mechanisms. This is particular for Chinese text, as even single word aspect target can be decomposed to up to three sequences of representation. Figure 5.1 presents a graphical illustration of ATSM-F in late fusion. In English, however, our model only applies to cases where aspect target contains multiple words. We will illustrate each of the three steps below.

5.4 Adaptive Embedding Learning

5.4.1 Sentence Sequence Learning

Sequential information is crucial in determining aspect term sentiment polarity. For example, there are two sentences: “*The movie supposed to be amazing but I find it just so-so.*” and “*The movie supposed to be just so so but I find it amazing.*” These two sentences have exactly the same words but arranged in a different order. This results in the opposite sentiment polarity of the aspect target “*The movie*”. In order to extract sentence sequential information like the above, we use an LSTM to encode the sentence sequential information. The output of the LSTM is a sequence of cell hidden outputs which has the same length of the sentence. Mathematically, a sentence and its corresponding LSTM sequence output are denoted as $\{w_1, w_2, \dots, w_j, \dots, w_{n-1}, w_n\}$ and $\{h_1, h_2, \dots, h_j, \dots, h_{n-1}, h_n\}$, respectively, where $w_n \in \mathbb{R}^{1 \times d}$ and $h_n \in \mathbb{R}^{1 \times d_{lstm}}$.

5.4.2 Aspect Target Unit Learning

As we discussed before, the meaning of aspect target word may be shifted due to the sentence context, such as the example of “*red apple*”. Thus, we decide to embed the intra-sentence contexts to each unit in the aspect target. To this end, we employ an attention mechanism to realize the learning. As we know from Bahdanau et al. [32], the attention mechanism can be understood as a weighted memory of lower-level elements. Conceptually, the output attention vector extracts the correlation between query (in our case which is the unit in aspect target) with each element. In our model, we compute an attention for each unit in aspect target with LSTM sequence hidden output from sentence sequence learning and name it adaptive vector. Thus, the adaptive vector extracts the most relevant correlation with intra-sentence context.

Specifically, for an aspect target unit u_i and its word embedding $v_i \in \mathbb{R}^{1 \times d}$ in a sentence of length n , its adaptive vector $V_{adapt} \in \mathbb{R}^{1 \times (d+d_{lstm})}$ is given as below:

$$V_{adapt} = \sum_{j=1}^n \alpha_j \cdot \begin{bmatrix} v_i \\ h_j \end{bmatrix} \quad (5.1)$$

where $h_j \in \mathbb{R}^{1 \times d_{lstm}}$ is the j th output from LSTM hidden output sequence. α_j is the weight for the j th memory in the sentence and $\sum_{j=1}^n \alpha_j = 1$. It depicts how much the semantic influence of the j th unit imposed on the aspect target unit u_i . It is a weight computed from *softmax* below:

$$\alpha_j = \frac{e^{g_j}}{\sum_{m=1}^n e^{g_m}} \quad (5.2)$$

where g_j is a score obtained from a feed-forward neural network attention model and its formula is given as:

$$g_j = \tanh(W_j \cdot \begin{bmatrix} v_i \\ h_j \end{bmatrix} + b) \quad (5.3)$$

where $W \in \mathbb{R}^{(d+d_{lstm}) \times 1}$ and $b \in \mathbb{R}^{1 \times 1}$.

We compute the adaptive vector for each unit in aspect target. In the end, we will get as many adaptive vectors as the number of aspect target units. Each of these adaptive vectors concentrates the influence that sentence context imposed on aspect target, namely it enriches the semantic meaning of aspect target by extracting correlation from intra-sentence context. For instance, the meaning of “*apple*” in our previous example.

5.5 Sequence Learning of Aspect Target

Having obtained the adaptive vector of each aspect target unit, we will extract the sequential information in the aspect target sequence. We find that sequential information within aspect target sequence is crucial in representing the meaning of an aspect target. Recall the previous example of “*red apple*”. Only by connecting “*red*” and “*apple*” will we have a complete impression of the new iPhone 7 in red coating. Isolating the two aspect words will be harmful. Therefore, we employ the second LSTM neural network [144] to encode such sequential information.

Specifically, we concatenate the adaptive vector of each aspect target unit to form an aspect target sequence. This sequence will be fed to an LSTM as input. In the end, we take out the hidden output H_L from the last LSTM cell as the representation of aspect target sequence.

5.6 Fusion of Multi-Granularity Representation

Unlike Latin languages such as English, Chinese written language is a type of pictogram, where its primitive forms symbolize certain meanings, such as characters ‘日 (sun)’ and ‘月 (moon)’. With time went by, more complex meanings need to be represented in text. Thus, certain simple characters cluster together to form complex characters. For instance, ‘明 (shining)’ composed of two sub-element characters ‘日 (sun)’ and ‘月 (moon)’. The semantic relation between is both ‘日 (sun)’ and ‘月 (moon)’ emit light and bring brightness. Simple characters like ‘日’ and ‘月’ were thus called ‘radical’ when they appear in constructing complex characters. In order

to represent abstract meanings, certain complex characters were clustered to form words. For instance, word ‘明星 (celebrity)’ composed of character ‘明 (shining)’ and ‘星 (star)’. Celebrities are shining stars in a sense.

Due to the above reasons, we understand modern Chinese text can be represented at three different granularities: radical, character and word. Inspired by [145], we represent Chinese text at three different granularities in our model and study the possible outcomes by fusion any of them.

In order to fit Chinese text into our deep learning framework, we represent Chinese text with embedding vectors. Particularly, we use the skip-gram model [115] to learn the embedding vectors at different granularities. Our training corpus contains about 8 million Chinese words, which is equal to 38 million Chinese characters or 150 million Chinese radicals. For word embedding vectors, we conduct word segmentation on the corpus using ICTCLAS [71] segmenter and then train with the skip-gram model. For character embedding vectors, we split each word in the corpus to individual characters, in which we still keep the order of characters. For radical embedding vectors, we decompose each character into radicals and concatenate them in the order from left to right. The decomposition was based on a Chinese character-radical look-up table we built using the Chinese character parser ‘HanziCraft¹’.

We design two fusion mechanisms (early and late) to merge three granularities. Early fusion concatenates different granularities of each aspect unit before aspect target sequence learning. Specifically, each aspect target word was represented by a concatenation of its sub-granular representations before it was sent to aspect target sequence learning. The output from aspect target sequence learning step will be fed to a *softmax* classifier.

Late fusion concatenates different granularities after aspect target sequence learning. Thus, for each granularity, an aspect target sequence representation will be obtained first. These representations will be concatenated and fed to a *softmax* classifier. Figure 5.1 presents a graphical illustration of ATSM-F in late fusion.

¹<http://hanzicraft.com>

5.6.1 Early Fusion

We have already proposed the fundamental model ATSM-S for ABSA. However, the performance of the model should largely depend on the representation of the text because the embedding vectors are the initial input to the model. To this end, we plan to incorporate the multi-level representation of Chinese text. ATSM-S emphasizes the role of word level of representation. We incorporate multi-level representation for aspect word to further improve the accuracy of aspect words. Instead of using only word level representation in ATSM-S, we explore using either two or three levels of representation, namely radical, character and word level.

Specifically for each sentence, we construct three types of sentence strings: a word string, a character string, and a radical string. In each of the string, aspect words are decomposed into the corresponding level. For each unit in the decomposed string of aspect words, it will learn an attention vector between the whole sentence string. For example, given an aspect word ‘工艺 (craftsmanship)’. One word attention vector will be learned from the word string. Two character attention vectors will be learned from the character string because the aspect word contains two characters, ‘工’ and ‘艺’. Three radical attention vectors will be learned due to the aspect word can be decomposed into three radicals, ‘工’, ‘艹’ and ‘乙’. Then, we compute an average attention vector for each representation level. Three resulting average attention vectors will be concatenated and will be treated as a fusion of multi-level representation. As this fusion is a feature level fusion for aspect term, we call this fusion the early fusion. The fusion attention vector will be fed to an LSTM like in ATSM-S. The final output from the LSTM will be fed to a *softmax* classifier. Graphical illustration is in Figure 5.2.

5.6.2 Late Fusion

Unlike the early fusion, where the fusion takes place at the feature level, in late fusion, the fusion of multi-level representation happens at classification step.

In late fusion, our ATSM-S is used intact at three different levels independently. As shown in Figure 5.2(b), the green break line box stands for ATSM-S working at

(a) Early fusion. (ATSM-S w/o stands for ATSM-S without sequence learning of aspect target)

(b) Late fusion.

Figure 5.2: Fusion mechanisms.

Table 5.1: Metadata of Chinese dataset

	Notebook	Car	Camera	Phone	Overall
Positive	417	886	1558	1713	4574
Negative	206	286	673	843	2008
Percentage of multi-word aspect	38.20	36.95	44.55	40.49	41.02

the word level. While the purple box stands for working at the character level and the blue box stands for working at the radical level. We take out the last LSTM hidden output from each level and concatenate them. The resulting concatenated vector will be fed to a softmax classifier.

Late fusion differs from early fusion in assuming that semantics within a sentence should be unified at the representation level. In other words, the semantics of aspect terms at a single level can hardly help extract semantics at other representation levels. Thus, in late fusion, the ATSM-S works merely on one level at a time and combines at the final classification step only.

5.7 Evaluation

In this section, we present our evaluations in three steps. The first step will conduct experimental evaluations of various methods to model aspect target sequence, as well as adaptive embedding learning. The second step will compare the proposed ATSM-S with the state of the art. The last step will evaluate the improvement by fusions of granularities. We used Tensorflow and Keras to implement our model. All models used AdagradOptimizer with a learning rate of 0.1 and an L2-norm regularizer of 0.01. Each mini-batch contains 50 samples. We report the average testing results of each model for 50 epochs in 5-fold cross validation.

5.7.1 Datasets

We used four Chinese datasets in four different domains for evaluation. They [146] contains reviews in four domains: notebook, car, camera, and phone. Aspect targets were originally tagged by [147]. Then, we manually labeled the sentiment polarity towards each aspect target as either positive or negative. The metadata of the dataset

was displayed in Table 5.1. English dataset used in our experiments was a subset from the SemEval2014 [148], which contains reviews from two different domains: restaurant and laptop. We select the reviews which contain multi-word aspect target only and results in a subset of 2309 reviews (30% of the original dataset).

5.7.2 Comparison Methods

Three types of baselines were included in our experiments. The first type includes the self-variants of ATSM-S, which examines the validity of each module of our model. The second type is the state-of-the-art methods in ABSA task, which tests the overall performance. The last type explores how the fusion of multi-grained representation of Chinese will affect the ABSA task.

5.7.2.1 Variants of ATSM-S

As there are two major modules of ATSM-S, namely adaptive embedding learning and sequence learning of aspect target, we design different variants for either of the modules to validate its superiority. **ATSM-v1** and **ATSM-v2** were designed to examine adaptive embedding module. **ATSM-v3**, **ATSM-v4** and **ATSM-v5** were designed to examine the module of sequence learning of aspect target.

- (i) **ATSM-v1**: The first variant of ATSM-S. It eliminates the sentence sequence learning step in ATSM-S. In the following steps, it replaces sentence level LSTM hidden state outputs with initial word embeddings.
- (ii) **ATSM-v2**: The second variant of ATSM-S. It removes the adaptive embedding learning module. Instead, it takes the sentence level LSTM hidden state outputs of each aspect target word as the input to aspect target sequence learning module.
- (iii) **ATSM-v3**: The third variant of ATSM-S. It replaces the module of aspect target sequence learning with an average of aspect target word embeddings. It doesn't extract the aspect target sequence information.

- (iv) **ATSM-v4**: The fourth variant of ATSM-S. It opts for different modeling of the aspect target sequence. Specifically, it replaces the LSTM at aspect target sequence level in ATSM-S with a CNN.
- (v) **ATSM-v5**: The last variant of ATSM-S. Unlike ATSM-v4 which models the aspect target sequence with a CNN, ATSM-v5 concatenates their embeddings and feeds to a nonlinear neural layer.
- (vi) **ATSM-S**: There are three sub-categories of this type. The first one is ATSM-S working on the word level. The other two are ATSM-S models working on character and radical level, respectively. These three variants do not have any fusion of representation levels and, hence, serve as baselines towards our fusion mechanism.

5.7.2.2 State of the art

We include several state-of-the-art methods: SVM, LSTM, Bi-LSTM, TD-LSTM, TC-LSTM, MemNets and ATSM-S.

- (i) **SVM**: SVM classifier trained on the surface and parse features, such as unigram, bigram, POS tags. Aspect target features were concatenated to sentence features.
- (ii) **LSTM**: The typical sequence modeling method that unveils the sequential information from the head to the tail of the sentence. It does not pay special attention to aspect term in the sentence. For long sentences, this method leverages more on the ending words in the sentence than the beginning words. Thus, for the case when aspect term appears in the head of the sentence, it may not work well.
- (iii) **Bi-LSTM**: It adds a reverse sequential learning step to LSTM. Bi-LSTM models both head to tail and tail to head sequential information, however, it does not distinguish the aspect term with context words in ABSA.

Table 5.2: Variants of ATSM-S on Chinese datasets at word level.

	Notebook		Car		Camera		Phone		Overall	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
ATSM-v1	69.98	62.60	80.88	55.09	78.27	66.81	80.83	68.51	81.95	77.05
ATSM-v2	66.94	40.58	75.59	42.29	69.94	47.08	67.72	42.58	70.24	45.64
ATSM-v3	74.15	62.04	80.71	57.24	78.09	69.49	81.65	71.59	81.89	76.47
ATSM-v4	74.80	60.00	82.94	59.43	82.34	69.86	84.11	73.24	85.76	80.84
ATSM-v5	73.35	58.67	79.61	56.65	78.31	68.33	80.56	70.03	82.42	77.84
ATSM-S	75.59	60.09	82.94	64.18	82.88	72.50	84.86	75.35	85.95	80.13

- (iv) **TD-LSTM**: Instead of attending to the complete length of a sentence like LSTM, TD-LSTM [45] used a forward and a backward sequence that ends immediately after including aspect term. It extracts sentence semantics prior to and after aspect term separately.
- (v) **TC-LSTM**: In addition to TD-LSTM, TC-LSTM appended the sentence word embedding with aspect target embedding. It hopes this procedure could explicitly capture the interaction between aspect word and sentence context word. Nevertheless, this method treated the sequential information from aspect target sequence and sentence word sequence with equal importance. They did not model the aspect target sequence.
- (vi) **MemNet**: This method took out the aspect word and looked for correlation with sentence context words. The problem of this method is it did not use the sequential information within the aspect target sequence. In our experiments on both English and Chinese datasets, we conducted various experiments using hop number from one to nine of this model and reported the best results.

5.7.2.3 Fusion comparison

- (i) **ATSM-F**: Based on ATSM-S, it fuses not only three representation granularities but also any two representation granularities, in both early and late manner. There are 11 different settings in this experiment. It evaluates whether fusion will improve from single granularity and which combination benefits the final result most.

5.7.3 Result Analysis

5.7.3.1 Self Comparison

In this section, we compare different variants of ATSM-S with experiments on Chinese datasets. The experimental results were shown in Table 5.2.

It can be observed that ATSM-S achieves the highest accuracy in all the datasets and highest F-score in three datasets. It generally demonstrates the validity of our model design. In order to elaborate more details, we conduct the comparisons with model variants.

ATSM-v1 differs from ATSM-S in that the former omits sentence sequential information. The decrease of performance from ATSM-v1 proves that ATSM-S has successfully encoded sentence sequence. Even if the sentence sequential information is correctly learned, the overall performance cannot be guaranteed. This is illustrated by ATSM-v2, which encodes the sentence sequence but does not learn adaptive embedding. Since ATSM-S learns adaptive embedding on top of ATSM-v2, a more accurate aspect target representation is learned and, hence, contributes to the final performance. ATSM-v3, -v4 and -v5 distinguish with each other in the way they model aspect target sequence. -v3 takes the average of aspect target word embeddings, which ignores aspect target sequential information. -v4 models the aspect target sequence with a CNN. -v5 models the sequence with the middle layer from an MLP. In comparison, ATSM-S models the sequence with an LSTM. From the table, we can conclude that LSTM achieves the best results compared with other variants. It further proves our assumption that the aspect target sequential information plays a significant role in ABSA task.

5.7.3.2 Peer Comparison

From Table 5.3, we can see that ATSM-S beats other state-of-the-art methods by around 1-4% in all four datasets and in one overall dataset.

The first reason why ATSM-S wins over other methods is that we explicitly learned the adaptive meaning of each aspect target unit. The adaptive embedding of each

Table 5.3: Accuracy and Macro-F1 results on Chinese datasets at word level.

	Notebook		Car		Camera		Phone		Overall	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
SVM	66.92	40.09	75.60	43.04	69.83	41.11	67.02	40.11	69.49	41.00
LSTM	74.63	62.32	81.99	58.83	78.31	68.72	81.38	72.13	82.71	78.28
Bi-LSTM	74.15	63.09	81.82	56.42	78.35	69.35	81.45	70.42	82.22	76.93
TD-LSTM	67.10	40.58	76.53	46.47	70.48	51.46	69.17	52.40	70.56	51.72
TC-LSTM	68.39	50.57	76.19	50.99	70.88	54.79	69.88	54.26	70.66	53.60
MemNet	69.10	53.51	75.55	51.01	70.59	55.13	70.29	55.93	72.86	55.99
ATSM-S	75.59	60.09	82.94	64.18	82.88	72.50	84.86	75.35	85.95	80.13

Table 5.4: Accuracy and Macro-F1 results on single-word/multi-word aspect target subset from SemEval2014.

	ATSM-S (word)		MemNet		TC-LSTM		TD-LSTM		Bi-LSTM	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Multi-word aspect English dataset	65.37	36.54	58.54	42.16	63.58	43.87	63.48	47.16	62.19	45.02
Single-word aspect English dataset	75.39	54.12	67.83	52.70	59.33	49.58	68.38	52.95	72.80	54.35

aspect target unit not only carries semantics from general word embedding but also encodes semantics within the sentence. In comparison, the baseline model ATSM-v2 eliminates sentence sequence learning step and, hence, results in a poor adaptive embedding.

We believe the second reason is that we explicitly modeled the aspect target sequence. Other state-of-the-art works either ignored the aspect target sequence [30, 33] or treated aspect target sequence as equal importance as sentence sequence [45]. Both of the approaches did not render enough emphasis on the aspect target sequence. To validate its importance, we designed the second baseline variant of ATSM-S, which is ATSM-v3. It differs from ATSM-S only in ignoring target sequence information. The sharp decrease of performance from ATSM-S to ATSM-v3 validated our assumption.

Differences between ATSM-S with the popular attention model where the aspect is embedded by an LSTM layer are two-fold. Firstly, ATSM-S specifically encodes aspect target sequential information. Whereas attention model treated aspect target as an averaged embedding vectors. Secondly, aspect target sequential information was given higher importance than sentence sequential information in ATSM-S. Whereas attention model treated the two sequences of equal importance.

Since ATSM-S specializes in modeling aspect target sequence, we conduct further experiments to test whether it is language independent. Thus, we removed reviews from English SemEval2014 dataset that had single-word aspect target only (e.g., pasta) and collected the remaining reviews that all had multi-word aspect target (e.g., build quality) to form a multi-word aspect target subset. Meanwhile, we collected the removed reviews to form a single-word aspect target subset.

Table 5.4 shows the experimental results on these two subsets in comparison with the top few state-of-the-art works, namely MemNet, TC-LSTM, TD-LSTM, and Bi-LSTM. In the single-word case, the proposed ATSM-S achieved the highest accuracy. It is beyond our expectation because the module of sequence learning of aspect target from ATSM-S would not work on single-word aspect target. On the other hand, it validates the contribution of the adaptive embedding learning module, which learns an accurate presentation of aspect target. In the other case, the table shows ATSM-S is the best at predicting multi-word aspect sentiment polarity in English dataset than the state of the art. The main reason is that our model explicitly learns adaptive embedding and aspect target sequence, where the latter is crucial. A visual analysis is provided in the next section.

Figure 5.3: Visual attention weights of each word in the example. (a) is from ATSM-S. (b) is from baseline model.

5.7.4 Visual Case Study

We visualize the difference between ATSM-S and a typical baseline model (MemNet) via a case study from English SemEval 2014 dataset.

We plot the heatmap of attention weights in Fig. 5.3. The deeper the color is, the heavier weight the word has. Our ATSM-S has two heatmaps due to we explicitly learn

adaptive embedding for each aspect target word (‘Korean’ and ‘dishes’). Whereas MemNet only has one, because it averages the word embeddings of aspect target and learns a sentence-level attention. It is apparent that either of our aspect unit adaptive embedding captures a key opinion word in the sentence, which are ‘affordable’ and ‘yummy’. In the later step of aspect target sequence learning, both of the opinion words will be captured and reflected in our final model output. Nevertheless, the heatmap from MemNet is the final model output, which unfortunately misses a crucial part of the opinionated content. The case study provides an intuitive explanation of why our ATSM-S prevails.

5.7.5 Granularity and Fusion Analysis

In the last set of experiments, we evaluated if multiple granularities in Chinese text representation will improve the performance of our model further. As shown in Table 5.5, we performed ATSM-S at each of the three granularities as baselines. We also applied ATSM-F in both early fusion mode and late fusion mode. The ATSM-F in the late fusion of word and character level achieved high results in four out of five datasets. It beat ATSM-S in almost any single granularity situation (except word level on Car dataset. However, it is close to the performance of ATSM-S at the word level.), which proved that a fusion of multiple granularities promoted the sentiment inference over single granularity.

Generally, we could see ATSM-S at character level produces the top few results in all single granularity cases. However, we found that word level performed better than character level on notebook and car datasets, a deeper look into those two datasets revealed the possible cause of biased data distribution. After computing variances of experimental results for each dataset, we found that the average variance of Notebook and Car dataset is 1.7 times bigger than the average variance of all five datasets at the word level, and 1.29 times larger than the average at the character level. This indicates that our model performance in these two datasets was less robust than the other three datasets. Furthermore, we found that the number of unique aspect target in these two datasets were relatively higher compared with their dataset size. This

Table 5.5: Accuracy results of multi-granularity with and without fusion mechanisms. (W, C, R stands for word, character and radical level, respectively. + means a fusion operation.)

			Notebook	Car	Camera	Phone	Overall
ATSM-S		W	75.59	82.94	82.88	84.86	85.95
		C	74.32	81.56	87.98	88.34	88.50
		R	69.92	75.68	77.19	78.09	79.87
ATSM-F	Early Fusion	W+C	77.52	82.16	86.55	87.13	89.38
		W+R	68.38	76.61	77.73	78.29	83.64
		C+R	69.99	77.81	80.73	80.90	87.41
		W+C+R	69.99	77.55	78.76	78.91	84.94
	Late Fusion	W+C	73.67	82.93	88.30	88.46	89.33
		W+R	67.26	78.23	80.68	84.94	86.43
		C+R	67.58	79.00	87.63	88.14	88.50
		W+C+R	67.91	78.15	87.98	88.07	89.30

explains why our model did not generalize well on these two datasets. Moreover, due to their smaller size compared, we believe all the above caused the character level performed worse than word level. In comparison, in the other three datasets, whose variances were well below average, character level outperformed word level with an obvious advantage. This is consistent with our expectation, as working at character level will wipe out the negative effects brought by word segmentation. It also explains why ‘W+C’ achieved the top few results. Sentiment information from the character level is effectively extracted and properly maintained with the help of effective character embeddings and ATSM-S. Fused with the word level information, the character level sentiment information improved the overall performance. However, working at the radical level did not improve the performance much, if not exacerbating the situation. Thus, it drove us to analyze the reason. We studied the aspect target distribution for each of the three representation granularities with our experimental datasets as examples. As shown in Figure 5.4, we plot the percentage of token types (i.e, unique tokens) of three different granularities appearing less than 10 times in the whole dataset. It is obvious to find that the representation of character level largely reduces the percentage of token types occurring only once in the word level. That is to say, character level representation significantly reduces the data sparsity of rare words by decomposing the words into characters. This explains why character level representation could greatly improve from word level. Radical level, on the

Figure 5.4: Percentage of number of terms in corresponding to from 1 to 10 occurrences in three-level representation.

contrary, does not reduce much the percentage from character level. One possible reason could be the ineffectiveness of our radical embedding vectors. In training the radical embeddings, we did not distinguish the radicals by phoneme and morpheme. This may introduce errors to radical embeddings, as phonemes do not carry semantics. These radical embeddings could affect the final results drastically. That being said, the radical level representation is still comparable to other baseline models. It indicates the potential of introducing radical level representation in the task of Chinese ABSA.

We elaborate why ATSM-F in late fusion mode achieves the top few performance as below. Our fusion mechanisms experimented on possible combinations of three granularities extract multi-granular semantics in the task of ABSA. In comparison, late fusion has a flat structure, while early fusion has a hierarchical structure. Using a flat structure means the semantic encoded by each granularity is relatively independent. Whereas using a hierarchical structure breaks down the semantic flow along each granularity. Because it fuses the representations of multi-granularity for each aspect target word, semantics at character and radical level were cut off by the boundary of words.

5.8 Summary

In this chapter, we investigated the problem of aspect-level sentiment analysis from a new perspective, in which the aspect target sequence dominates the final result. Accordingly, we proposed ATSM-S, which prevailed the state of the art in multi-word aspect sentiment analysis on SemEval2014. Moreover, our model specifically catered to the multi-granularity representations of Chinese text. By designing a late fusion method, ATSM-F outperformed all state-of-the-art works in three Chinese review datasets. As one of the first attempts to exploiting multi-granularity nature of Chinese ABSA, this work paves the path for potentially wider application scenarios in Chinese natural language processing.

Chapter 6

Phonetic-enriched Text Representation

6.1 Introduction

While most literature addresses sentiment analysis in a language-independent approach, Chinese sentiment analysis, in fact, requires tackling language-dependent challenges due to its unique nature, including word segmentation [149], compositional analysis [42, 51, 43, 150, 145]. There are two main characteristics distinguishing Chinese from other languages. Firstly, it is a *pictogram* language [151], which means that symbols (called Hanzi) intrinsically carry meanings. Multiple symbols might form a new single symbol via geometric composition. The hieroglyphic nature of Chinese writing system differs from many Indo-European languages such as English or German. It has therefore inspired many works to explore the sub-word components (such as Chinese character and Chinese radicals) via a textual approach [150, 41, 42, 51, 43, 145]. The other research approach models the compositionality using the visual presence of the characters [52, 122] by the means of extracting visual features from bitmaps of Chinese characters to further improve the Chinese textual word embeddings.

The second characteristic of Chinese is that it is a language of deep phonemic orthography according to the orthographic depth hypothesis [34, 35]. In other words, it is hard to support the recognition of words involving the language phonology. Each symbol of the modern Chinese language can be phonetically transcribed into a romanized form, called Hanyu PinYin (or pinyin), consisting of an initial (optional),

a final, and the tone. More specifically, as a tonal language, one single syllable in modern Chinese can be pronounced with five different tones, i.e., 4 main tones and 1 neutral tone (shown later in Table 6.4). We would argue that this particular form of the Chinese language provides semantic cues complementary to its textual form as illustrated in Table 1.2. Despite its important role in the Chinese language, to the best of our knowledge, it has not yet been explored by existing work for NLP tasks of the Chinese Language.

In this work, we argue that the second factor of the Chinese language can play a vital role in Chinese natural language processing especially sentiment analysis. Particularly, to consider the deep phonemic orthography and intonation variety of the Chinese language, we propose two steps to learn Chinese phonetic information.

Firstly, we come up with two types of phonetic features. The first type extracted audio features from real audio clips. The second type learned pinyin token embeddings from a converted pinyin corpus. For each type of feature, we provide one version with intonation and one version without intonation.

Upon building the feature lookup table between each Chinese pinyin and its feature/embedding, we reach our second step, which is to design a DISA (disambiguate intonation for sentiment analysis) network that works on pinyin sequence and automatically decides the correct intonation for each pinyin. This step is crucial in disambiguating meanings and even the sentiment of Chinese characters. Specifically, inspired by [152], we employ a reinforcement network as the main structure for our DISA network. The actor network is a typical neural policy network, whose action is to choose one out of five intonations for each pinyin. The critic network is an LSTM sequence model, which learns the pinyin sentence sequence representation. The policy network is updated by a delayed reward when the sequence representation is built, while the critic network is updated by a sentiment class cross-entropy loss.

Motivated by the recent success of multimodal learning, we also incorporate textual and visual features with phonetic features. To the best of our knowledge, we are the first to consider the deep phonemic orthographic characteristic and intonation variation in a multimodal framework for the task of Chinese sentiment analysis.

The experimental results show that the proposed multimodal framework outperforms the state-of-the-art Chinese sentiment analysis method by a statistically significant margin. In summary, we make three main contributions in this chapter:

- We augment the representation of Chinese characters with phonetic cues.
- We introduce a reinforcement learning based framework, DISA, which jointly disambiguates intonations of Chinese characters and resolve the sentiment polarity classes of the sentence.
- We demonstrate the effectiveness of our framework on several benchmark datasets.

The remainder of this chapter is organized as follows: we first present a brief review of embedding features, sentiment analysis and Chinese phonetics; we then introduce our model and provides technical details; next, we describe the experimental results and present analytical discussions; finally, we conclude the chapter.

6.2 Model

In this section, we first present how features from textual and visual modalities were extracted. Next, we delve deep into the details of different types of phonetic features. Then, we introduce a DISA network which parses Chinese characters to their pronunciations with tones. Lastly, we demonstrate how we fuse features from three modalities for sentiment analysis.

6.2.1 Textual Embedding

As in most recent literature, textual word embedding vectors were treated as the fundamental representation of texts [115, 109, 153]. Firstly introduced by Bengio et al. [109], low-dimensional word embedding vectors learned a distributed representation for words. Compared with traditional n-gram word representations, they largely reduced the data sparsity problem and provided more friendly access towards neural networks. In 2013, Mikolov et al. [115] introduced the toolkit ‘Word2Vec’ which populated the application of word embedding vectors due to its fast learning time.

In the toolkit, two predictive-based word vectors, CBOW and Skip-gram, were proposed. They either predicted the target word from context or vice versa. Pennington et al. [153] developed ‘GloVe’ in 2014 which employed a count-based mechanism to embed word vectors. Following the convention, we used ‘GloVe’ character embeddings [153] of 128-dimension to represent text.

It is worth noting that we set the fundamental token of Chinese text as the character instead of the word for two reasons. Firstly, the character is designed to align against the audio feature. Audio features can only be extracted at the character level, as Chinese pronunciation is on each character. In the Chinese language, the fundamental phonetic unit which is semantically self-contained is at the character level. In English, however, the fundamental phonetic unit is at the word level (except some prefix/suffix syllables). Secondly, character-level processing can avoid the errors induced by the Chinese word segmentation. Although we used character GloVe embedding as our textual embedding, experimental comparisons were conducted with both CBOW [115] and Skip-gram embeddings.

6.2.2 Training Visual Features

Unlike the Latin language, the Chinese written language originated from pictograms. Afterward, simple symbols were combined to form complex symbols in order to express abstract meanings. For example, a geometric combination of three ‘木 (wood)’ creates a new character ‘森 (forest)’. This phenomenon gives rise to a compositional characteristic of Chinese text. Instead of a direct modeling of text compositionality using sub-word [41, 150] or sub-character [51, 42, 145] elements, we opt for a visual model. In particular, we constructed a convolutional auto-encoder (convAE) to extract visual features. Details of the convAE are listed in Table 6.1.

Following the convention in [154] and [52], we set the input of the model to a 60 by 60 bitmap for each of the Chinese characters and the output of the model to a dense vector with a dimension of 512. The model was trained using Adagrad optimizer on the reconstruction error between original bitmap and reconstructed bitmap. The loss

Table 6.1: Configuration of convAE for visual feature extraction.

Layer#	Layer Configuration
1	Convolution 1: kernel 5, stride 1
2	Convolution 2: kernel 4, stride 2
3	Convolution 3: kernel 5, stride 2
4	Convolution 4: kernel 4, stride 2
5	Convolution 5: kernel 5, stride 1
Feature	Extracted visual feature: (1,1,512)
6	Dense ReLu: (1,1,1024)
7	Dense ReLu: (1,1,2500)
8	Dense ReLu: (1,1,3600)
9	Reshape: (60,60,1)

Figure 6.1: Original input bitmaps (upper row) and reconstructed output bitmaps (lower row).

is given as:

$$\sum_{j=1}^L (|x_t - x_r| + (x_t - x_r)^2) \quad (6.1)$$

where L is the number of samples. x_t is the original input bitmap and x_r is the reconstructed output bitmap. An example of the original and reconstructed bitmaps is shown in Figure 6.1. After training the visual features, we obtained a lookup table where each Chinese character corresponds to a 512-dimensional feature vector.

6.2.3 Learning Phonetic Features

Written Chinese and spoken Chinese have several fundamental differences. To the best of our knowledge, all previous literature that worked on Chinese NLP ignored the significance of the audio channel. As cognitive science suggests, human communication depends not only on visual recognition but also on audio activation. This drove us to explore the mutual influence between the audio channel (pronunciation) and textual representation.

Popular Latin and Germanic languages such as Spanish, Portuguese, English etc. share two remarkable characteristics. Firstly, they have shallow phonemic orthogra-

phy¹. In other words, the pronunciation of a word is largely dependent on the text composition in such languages. One can almost infer the pronunciation of a word given its textual spelling. From this perspective, textual information can be interchangeable with phonetic information.

For instance, if the pronunciations of the English word ‘subject’ and ‘marineland’ were known, it is not hard to speculate the pronunciation of word ‘submarine’, because one can combine the pronunciation of ‘sub’ from ‘subject’ and ‘marine’ from ‘marineland’. This implies that phonetic information of these languages may not have additional information entropy than textual information. Secondly, intonation information is limited and implicit in these languages. Generally speaking, emphasis, ascending intonation and descending intonation are the major variations in these languages. Although they exerted great influence in sentiment polarity during communication, there is no apparent clue to infer such information only from the texts.

However, the Chinese language differs from the above-mentioned languages in several key aspects. Firstly, it is a language of deep phonemic orthography. One can hardly infer the pronunciation of the Chinese word/character from its textual writing. For example, the pronunciations of characters ‘日’ and ‘月’ are ‘rì’ and ‘yuè’, respectively. A combination of the two characters makes another character ‘明’ which pronounced ‘míng’. This characteristic motivates us to find how the pronunciation of Chinese can affect natural language understanding. Secondly, intonation information of Chinese is rich and explicit. In addition to emphasis, each Chinese character has one tone (out of five different tones), marked by diacritics explicitly. These intonations (tones) greatly affect the semantic and sentiment of Chinese characters and words, as in Table 1.2.

To this end, we found it was not trivial to explore how Chinese pronunciation can influence natural language understanding, especially sentiment analysis. In particular, we designed two approaches to learn phonetic information, namely feature extraction from audio signal and embedding vector learning from the textual corpus. For either of the above two approaches, we have two variations, namely with (Ex04, PW) or

¹en.wikipedia.org/wiki/Phonemic_orthography

Table 6.2: Illustration of 4 types of phonetic features: $a(x)$ stands for the extracted audio feature for pinyin ‘x’; $v(x)$ represents learned embedding vector for ‘x’; number 0 to 4 represents 5 diacritics.

Text		假设明天放假。
English		Suppose tomorrow is holiday.
Pinyin		Jiǎ Shè Míng Tiān Fàng Jià
Extracted from audio	Ex0	$a(Jia)$ $a(She)$ $a(Ming)$ $a(Tian)$ $a(Fang)$ $a(Jia)$
	Ex04	$a(Jiǎ)$ $a(Shè)$ $a(Míng)$ $a(Tiān)$ $a(Fàng)$ $a(Jià)$
Learned from corpus	PO	$v(Jia)$ $v(She)$ $v(Ming)$ $v(Tian)$ $v(Fang)$ $v(Jia)$
	PW	$v(Jia3)$ $v(She4)$ $v(Ming2)$ $v(Tian1)$ $v(Fang4)$ $v(Jia4)$

without (Ex0, PO) intonations. An illustration is shown in Table 6.2. Details of each type will be introduced in the following sections.

6.2.3.1 Extracted feature from audio clips (Ex0, Ex04)

The spoken system of modern Chinese is named ‘Hanyu Pinyin’, abbreviated to ‘pinyin’². It is the official romanization system for mandarin in mainland China [155]. The system includes four diacritics denoting four different tones plus one neutral tone. For each of the Chinese characters, it has one corresponding pinyin. This pinyin has five variations in tones (we treat the neutral tone as one special tone). The statistics of Chinese character and pinyin are listed in Table 6.3. It shows that the number of frequently used characters is bigger than the number of pinyins with or without tones. This suggests that certain Chinese characters share the same pinyin and further implies that the one-hot dimensionality will reduce if pinyin was used to represent text.

In order to extract phonetic features, for each tone of each pinyin, we collected an audio clip which recorded a female’s pronunciation of that pinyin (with tone) from a language learning resource³. Each audio clip lasts around one second with a standard pronunciation of one pinyin with tone. The quality of these clips was validated by two native speakers. Next, we used openSMILE [156] to extract phonetic features on each of the obtained pinyin-tone audio clip. Audio features are extracted at 30 Hz frame-rate and a sliding window of 20 ms. They consist of a total number of 39

²iso.org/standard/13682.html

³<http://chinese.yabla.com> – This resource has only four tones for each pinyin, which does not have the neutral tone pronunciation. To obtain the neutral tone feature, we compute the arithmetic mean of the features of the other four tones.

Table 6.3: Statistics of Chinese characters and ‘Hanyu Pinyin’

	Pinyin		Textual Character
	w/o tones	w/ tones	
Number of tokens	374	1870	3500

low-level descriptors (LLD) and their statistics, e.g., MFCC, root quadratic mean, etc.

After obtaining features for each of the pinyin-tone clip, we obtained an $m \times 39$ dimensional matrix for each clip, where m depends on the length of clip and 39 is the number of features. To regulate the feature representation for each clip, we conducted singular value decomposition (SVD) on the matrices to reduce them to 39-dimensional vectors, where we extracted the vector with the singular values. In the end, high dimensional feature matrices of each pinyin clip were transformed into a dense feature vector of 39 dimensions. A lookup table between pinyin and audio feature vector is constructed accordingly.

In particular, we prepared two sets of extracted phonetic features. The first type comes with tone, which is the feature we obtained from the above processing. We denote it as **Ex04**, where ‘Ex’ stands for extracted features and ‘04’ stands for having one tone from 0 to 4 (we represent neutral tone as 0 and the first to the fourth tone as 1 to 4 respectively). The second type removed the variations of tones, in which we take the arithmetic mean of five features from five tones of each pinyin. We denote it as **Ex0**, where ‘0’ stands for no tone. In the second type of feature, pinyins with different tones will have the same phonetic features, even though they may mean different meanings.

6.2.3.2 Learned feature from pinyin corpus (PO, PW)

Instead of collecting audio clips for each pinyin and extracting audio features, we directly represent Chinese characters with pinyin tokens, as shown in Table 6.2. Specifically, we convert each Chinese character in a textual corpus to its pinyin. The original corpus which was represented by a sequence of Chinese characters was converted to a phonetic corpus which was represented by a sequence of pinyin tokens.

In the phonetic corpus, contextual semantics were still maintained as in textual corpus. This is achieved with the help of online parser⁴, which parse Chinese characters to their pinyins. It should be pointed out that 3.49% of the common 3500 Chinese characters (around 122 characters) have multiple pinyins⁵, namely ‘duo yin zi’(heteronym). Although the parser claimed its support to heteronym, we took the most statistically-possible pinyin prediction of each heteronym.

We did not disambiguate various heteronyms particularly, as this is not the major assumption we try to argue in this work. However, it could be a direction worth working on in the future. The DISA provides two modes in its conversion from Character to pinyin, one with tone and the other without tone.

For the mode without tone, Chinese characters will be converted to pinyin without tones only. Examples are the tokens shown in the row of PO in Table 6.2, where **PO** stands for Pinyin w/o tones. Afterward, we train 128-dimension pinyin token embedding vectors using conventional ‘GloVe’ character embeddings [153]. A lookup table between pinyin without intonation (**PO**) and embedding vector is constructed accordingly. Pinyins that have the same pronunciation but different intonations will share the same GloVe embedding vector, such as *Jiǎ* and *Jià* in Table 6.2.

For the mode with tone, Chinese characters will be converted to pinyin plus a number suggesting the tone. Examples are the tokens shown in the row of PW in Table 6.2, where **PW** stands for Pinyin w/ tones. We use number 1 to 4 to represent four diacritics and number 0 to represent the neutral tone. Similarly, 128-dimension ‘GloVe’ pinyin embedding vectors were trained.

In sum, we have four types of phonetic features, namely **Ex04**, **PW**, **E0** and **PO**. **PO** distinguishes from **PW** in removing intonations. Two of them (**Ex04**, **PW**) distinguish from others by having intonations. It is expected to have one question that how would one know the correct intonation of pinyins given their textual characters. Although the online parser can give its statistical guess, the performance and robustness can not be evaluated and guaranteed. To address this problem, we design

⁴github.com/mozillazg/python-pinyin

⁵yyxx.sdu.edu.cn/chinese/new/content/1/04/main04-03.htm

Figure 6.2: DISA model structure for tone selection. C_m stands for the m th Chinese character in a sentence. P_m denotes the pinyin for m th character without the tones. P_{mn} represents the pinyin for m th character with its n th tone. F_{mn} is the feature/embedding vector for P_{mn} .

a parser network with a reinforcement learning model to learn the correct intonation of each pinyin. Details will be presented in the following section.

6.2.4 DISA

6.2.4.1 Overview

This DISA network takes a sentence of Chinese characters as input. It firstly converts each character to its corresponding pinyin (without tones) through a lookup operation. Then the pinyin sequence will be fed to an **actor-critic network**. For each pinyin (time step), a policy network will randomly sample one out of five actions, where each action denotes a tone. Then a feature/embedding of this specific pinyin with tone is retrieved from a feature lookup module.

During the exploration stage, the action will be randomly sampled. During exploitation and prediction stages, the action will be the one with maximum probability given the policy. This feature/embedding sequence will then be fed to an LSTM network. Hidden states from the LSTM will pass back to the policy network for guiding action selection. The final hidden state of the LSTM network will be fed to a softmax classifier to obtain a sentence sentiment class distribution. A log probability of ground-truth label will be treated as a delayed reward to tune the policy network. Finally, a cross entropy loss will be computed against the obtained sentiment class distribution to tune the critic network. A graphical description is shown in Figure 6.2, followed by details below.

State: For the environment, we used an LSTM to simulate the value function (detailed later). The input to this LSTM is the sequence of feature/embedding retrieved from the lookup module (detailed later), namely $x_1, x_2, \dots, x_t, \dots, x_L$, where x_t is the feature for the t th pinyin in the sentence. The mathematical representations of the LSTM cell are as follows:

$$\begin{aligned}
 f_t &= \sigma(W_f[x_t, h_{t-1}] + b_f) \\
 I_t &= \sigma(W_I[x_t, h_{t-1}] + b_I) \\
 \tilde{C}_t &= \tanh(W_C[x_t, h_{t-1}] + b_C) \\
 C_t &= f_t * C_{t-1} + I_t * \tilde{C}_t \\
 o_t &= \sigma(W_o[x_t, h_{t-1}] + b_o) \\
 h_t &= o_t * \tanh(C_t)
 \end{aligned} \tag{6.2}$$

where f_t , I_t and o_t are the forget gate, input gate and output gate, respectively. W_f, W_I, W_o, b_f, b_I and b_o are the weight matrix and bias scalar for each gate. C_t is the cell state and h_t is the hidden state output.

The state of the environment is defined as:

$$S_t = [x_t \oplus h_{t-1} \oplus C_{t-1}] \tag{6.3}$$

where $\oplus$ is a concatenation (same below). As shown in Formula 6.3, the state is determined by the current feature input, the last LSTM hidden output and the last LSTM cell memory.

Action: There are five actions in our environment, representing five different tones. An example is shown in Table 6.4. If different action was selected, then the corresponding intonation will be activated. Relevant phonetic features will then be selected, as introduced in Section 6.2.4.3. The action policy was implemented by a typical feedforward neural network. Specifically, for a policy $\pi(a_t | S_t)$ at time t ,

$$\pi(a_t | S_t) = \tanh(W \cdot S_t + b) \tag{6.4}$$

where W and b are the weight matrix and bias scalar. a_t is the action at time t . During exploration of training, action will be randomly selected out of the above five.

Table 6.4: Actions in DISA network and meanings.

Action	0	1	2	3	4
Intonation	Neutral	˘	˙	˚	˛
Example	a	ā	á	ǎ	à

During exploitation of training and testing, the action with the maximum probability will be selected.

Reward: The reward is computed by the end of each sentence when the state/action trajectory comes to the terminal (delayed reward). After the feature/embedding lookup module, the feature sequence is fed to the LSTM critic network. A sentence sentiment class distribution is computed as:

$$distr = \sigma(W_{softmax} \cdot h_L + b_{softmax}) \quad (6.5)$$

where $W_{softmax}$ and $b_{softmax}$ are weight matrix and bias scalar from the softmax layer. h_L is the last hidden state output from the LSTM critic network. $distr^{1 \times X}$ is the probability distribution of sentiment classes for the sentence. X is the number of sentiment class. The reward (R) is defined as:

$$R = \log(P(ground \mid sent)) \quad (6.6)$$

where $P(ground \mid sent)$ stands for the probability of the ground-truth label of the sentence given the distribution in Eq. 6.5.

6.2.4.2 Actor: policy network

As shown in the ‘Action’ above, the policy network random guesses actions during the exploration stage in training. It will be updated when a sentence input is fully traversed. Given the reward obtained from Eq. 6.6, we used gradient descent method to optimize the policy network [157]. In other words, we want to maximize:

Figure 6.3: An example of fused character feature/embedding lookup, where T, P, V represent features/embeddings from corresponding modality. In the case of single modality or bi-modality, relevant lookup table is constructed accordingly.

$$\begin{aligned}
 J(\theta) &= E_{\pi}[R(S_1, a_1, S_2, a_2, \dots, S_L, a_L)] \\
 &= \sum_1^L p(S_1) \prod_t \pi_{\theta}(a_t | S_t) p(S_{t+1} | S_t, a_t) R_L \\
 &= \sum_1^L \prod_t \pi_{\theta}(a_t | S_t) R_L
 \end{aligned} \tag{6.7}$$

Using the likelihood ratio (or REINFORCE [158] trick) to estimate policy gradient, the gradient can be transformed to:

$$\nabla_{\theta} J(\theta) = \sum_{t=1}^L R_L \nabla_{\theta} \log \pi_{\theta}(a_t | S_t) \tag{6.8}$$

6.2.4.3 Feature/embedding lookup

Recall that we have selected actions from actor network, where each action denotes a tone for that pinyin, the function of this feature/embedding lookup module is to retrieve the correct features of that specific pinyin with tone. Prior to the policy network, we have collected phonetic features from five different tones of each pinyin and order them from neutral tone feature to the fourth tone feature. The neutral tone to the fourth tone feature can be retrieved individually by index ID number 0 to 4.

When an action is selected from the actor network, for example, action 4 was selected for pinyin P_1 , this lookup module will find the fourth phonetic feature (index ID 4) of this pinyin, namely F_{14} and pass it to the LSTM critic network as the input x_t in Eq. 6.2.

6.2.4.4 Critic: sentence model and loss computation

Introduced in the **State** before, the critic network was essentially a sentence encoding model by an LSTM. We used gradient descent method to update the critic network with the cross-entropy loss defined as:

$$L = - \sum_{\forall sent} P(ground \mid sent) \log(P(pred \mid sent)) \quad (6.9)$$

where $P(ground \mid sent)$ and $P(pred \mid sent)$ are the ground truth and predicted probability in the Eq. 6.5, respectively.

6.2.5 Fusion of Modalities

In the context of the Chinese language, textual embeddings have been applied in various tasks and proved its effectiveness in encoding semantics or sentiment [41, 51, 42, 43, 150]. Recently, visual features pushed the performance of textual embedding further via a multimodal fusion [52, 122]. This is achieved due to the effective modeling of compositionality of Chinese characters by the visual features. In this work, we hypothesize that the use of phonetic features along with textual and visual can improve the performance. Thus, we introduced the following fusion method that fits with our DISA network, as in Figure 6.2.

- Each Chinese character is represented by a concatenation of three segments. Each segment represents one modality, see below:

$$char = [emb_T \oplus emb_P \oplus emb_V] \quad (6.10)$$

where $char$ is character representation. emb_T, emb_P, emb_V are embeddings from text, phoneme and vision, respectively.

There are other complex fusion methods available in the literature [159], however, we did not use them in our work for three reasons. (1) Fusion through concatenation is one proven effective method [160, 161, 122], and (2) it has the added benefit of simplicity, that allowing for the emphasis (contributions) of the system to remain with

the features themselves. (3) The designed fusion needs to fit in with our reinforcement model framework. Fusion methods as in [52] and [159] impose obstacles in the implementation with actor-critic model. Thus, we used the above-introduced fusion method, an example of a fused feature/embedding lookup table is shown in Fig. 6.3.

6.3 Experiments and Results

In this section, we start with introducing the experimental setup. Experiments were conducted in five steps. Firstly, we compare unimodal features. Secondly, we experiment on the possible fusion of modalities. Thirdly, we compare cross domain validation performance between our method with baselines. Next, we conduct ablation tests to validate the contribution of phonetic features. Lastly, we visualize different phonetic features/embeddings to understand how they improve the performance.

6.3.1 Experimental Setup

6.3.1.1 Datasets and features/embeddings

Datasets: We evaluate our method on five datasets: Weibo, It168, Chn2000, Review-4 and Review-5. The first three datasets consist of reviews extracted from micro-blog and review websites. The last two datasets contain reviews from [146], where Review-4 has reviews from computer and camera domains, and Review-5 contains reviews from car and cellphone domains. The experimental datasets are shown in Table 6.5.

Features/embeddings: For textual embeddings, we refer to the pretrained character embedding lookup table trained with Glove in Section 6.2.1. For phonetic experiments, we employ a pre-built tool called online codes⁶ on the datasets to convert text to pinyin without intonations (As we discussed in Section 6.2.3.2, this conversion achieves as high as 97% accuracy.). Ex0 and Ex04 features were extracted from audio files and stored as in Section 6.2.3.1. PO and PW embeddings were also pretrained on the same textual corpus for training textual embeddings. The corpus contains news of 8 million Chinese words, which is equal to 38 million Chinese characters. For visual

⁶github.com/mozillazg/python-pinyin

features, we refer to the lookup table to convert characters to visual features as in Section 6.2.2.

For experiments of multimodality, features from each individual modality were concatenated into a lookup table. Examples are shown in Fig. 6.3.

Table 6.5: Statistics of experimental datasets

	Weibo	It168	Chn2000	Review-4	Review-5
Positive	1900	560	600	1975	2599
Negative	1900	458	739	879	1129
Sum	3800	1018	1339	2854	3728

6.3.1.2 Setup and Baselines

Setup: We use TensorFlow and Keras to implement our model. All models use an Adam Optimizer with a learning rate of 0.001 and an L2-norm regularizer of 0.01. The dropout rate is 0.5. Each mini-batch contains 50 samples. We split each dataset to training, testing and development sets per the ratio 6:2:2. We report the result of the testing set whose corresponding development set performs the best after 30 epochs. The above parameters were set with the use of a grid search on the development data.

The training procedure for our DISA network is as follows. Firstly, we skip the policy network and directly train the LSTM critic network with the training objective as Eq. 6.9. Secondly, we fix the parameters of the LSTM critic network and train the policy network with the training objective as Eq. 6.8. Lastly, we co-train all the modules together until convergence. For the cases when no phonetic feature/embedding is involved, for example, pure textual or visual features, only the LSTM is trained and tested. The Glove was chosen as the textual embedding in our model due to its performance in Table 6.6.

DISA variants: We introduce below the variants of our DISA network. They differ in text representation features.

- (i) **DISA (P):** DISA network that used phonetic feature only, which is the concatenation of Ex04 and PW.

- (ii) **DISA (T+P)**: DISA network that uses the concatenation of textual embedding (GloVe) and phonetic feature (Ex04+PW).
- (iii) **DISA (P+V)**: DISA network that uses the concatenation of phonetic feature (Ex04+PW) and visual feature.
- (iv) **DISA (T+P+V)**: DISA network that uses the concatenation of textual embedding (GloVe), phonetic feature (Ex04+PW) and visual feature.

Baselines: Our proposed method is based on input features/embeddings of Chinese characters. In related works of Chinese textual embedding, all of them aimed at improving Chinese word embeddings, such as CWE [41], MGE [150]. Those who utilized visual features [52, 122] also aimed at word level. However, they cannot stand as fair baselines to our proposed model, as our model studies Chinese character embedding. There are two major reasons for studying at the character level. Firstly, the pinyin pronunciation system is designed for character level. Pinyin system does not have corresponding pronunciations to Chinese words. Secondly, the character level will bypass Chinese word segmentation operation which may induce errors. Conversely, using character level pronunciation to model word level pronunciation will incur sequence modeling issues. For instance, a Chinese word ‘你好’ is comprised of two characters, ‘你’ and ‘好’. For textual embedding, the word can be treated as one single unit by training a word embedding vector. For phonetic embedding, however, we cannot treat the word as one single unit from the perspective of pronunciation. The correct pronunciation of the word is a time sequence of character pronunciation of firstly ‘你’ and then ‘好’. If we work at the word level, we have to come up with a representation of the pronunciation of this word, such as an average of character phonetic features etc. To make a fair comparison, we compare with character level methods below:

- (i) **GloVe**: An unsupervised embedding learning algorithm based on co-occurrence (count). [153].

Table 6.6: Classification accuracy of unimodality in LSTM.

		Weibo	It168	Chn2000	Review-4	Review-5
	GloVe	75.39	81.82	84.54	87.46	86.94
	CBOW	72.39	78.75	81.18	85.11	84.71
	Skip-gram	75.05	80.13	78.04	86.23	86.21
	Visual	61.78	65.40	67.21	78.98	79.59
	charCBOW	71.54	80.83	82.82	86.90	85.19
	charSkipGram	71.86	82.10	81.63	85.21	84.84
	Hsentic	73.65	80.23	79.09	84.76	73.31
Phonetic features	DISA(Ex04)	67.28	84.69	78.18	81.88	83.38
	DISA(PW)	67.80	83.73	77.45	85.37	84.18
	DISA(P)	68.19	85.17	79.27	84.67	85.24

- (ii) **CBOW**: Continuous Bag-of-words model which places context words in the input layer and target word in the output layer [115].
- (iii) **Skip-gram**: The opposite of CBOW model, which predicts the contexts given the target word [115].
- (iv) **Visual**: Based on [52] and [154], a convolutional auto-encoder (convAE) is built to extract compositionality of Chinese characters through the visual channel.
- (v) **charCBOW**: Component-enhanced character embedding built on top of CBOW method by [51]. It delved into the radical components of Chinese characters and enriched the character representation with radical component.
- (vi) **charSkipGram**: The Skip-gram varaint of charCBOW.
- (vii) **Hsentic**: It proposed radical-based hierarchical embeddings for Chinese sentiment analysis. Character representations were specifically tuned for sentiment analysis [145].

6.3.2 Experiments on Unimodality

For textual embeddings, we have compared with state-of-the-art embedding methods including GloVe, skip-gram, CBOW, charCBOW, charSkipGram and Hsentic. As shown in Table 6.6, textual embeddings (GloVe) achieve the best performance among all three modalities in four datasets. This is due to the fact that they successfully

encoded the semantics and dependency between characters. We also find that char-CBOW and charSkipGram methods perform quite close to the original CBOW and Skip-gram methods. They perform slightly but not constantly better than their baselines. We conjecture this could be caused by the relatively small size of our training corpus compared to the original Chinese Wikipedia Dump training corpus. With the corpus size increased, all embedding methods are expected to have improved performance. It is without doubts, though, that the corpus we used still presents a fair platform for all methods to compare.

We also notice that visual feature achieve the worst performance among three modalities, which is within our expectation. As demonstrated in [52], pure visual features are not representative enough to obtain a comparable performance with the textual embedding. Last but not least, our methods with phonetic features perform better than the visual feature. Although visual features capture compositional information of Chinese characters, they fail to distinguish different meanings of characters that have the same writing but different tones. These tones could largely alter the sentiment of Chinese words and further affect the sentiment of a sentence.

For phonetic representation, three types of features were tested, namely EX04, PW and P (namely EX04+PW). The last one is the concatenation of the previous two. Our first observation is that phonetic features alone can hardly compete with textual embeddings. Although they beat textual embeddings in It168 dataset, they consistently fell behind textual embeddings. This is still within our expectation, as suggested by Tseng in [162], ‘Phonology and phonetics alone are insufficient in predicting the actual output of sentences’.

If we further refer to the Table 6.3, we can find that on average 2 to 3 characters share one same pinyin with tone. That means a pure phonetic representation may wipe out the 1 out of 2 or 3 (33%-50%) semantics from the text. This inevitably will reduce the possibility to correctly classify the sentiment.

As we can see each modality has its own capacity to encode semantics, it is expected to take advantage of the complementary information from multiple modalities for the sentiment analysis task. The results are shown in the next section.

Table 6.7: Classification accuracy of multimodality. (T and V represent textual and visual, respectively. + means the fusion operation. P is the concatenated phonetic feature of the one extracted from audio (Ex04) and pinyin w/ intonation (PW).)

	Weibo	It168	Chn2000	Review-4	Review-5
GloVe	75.39	81.82	84.54	87.46	86.94
Visual	61.78	65.40	67.21	78.98	79.59
charCBOW	71.54	80.83	82.82	86.90	85.19
charSkipGram	71.86	82.10	81.63	85.21	84.84
Hsentic	73.65	80.23	79.09	84.76	73.31
DISA(P)	68.19	85.17	79.27	84.67	85.24
DISA(T+P)	75.75	86.12	85.45	90.42	90.03
DISA(T+V)	73.79	85.65	83.27	89.37	88.70
DISA(P+V)	76.01	82.30	81.09	86.76	87.23
DISA(T+P+V)	74.32	77.99	78.18	87.63	89.49

6.3.3 Experiments on Fusion of Modalities

In this set of experiments, we evaluate the fusion of every possible combination of modalities. After extensive experimental trials, we summarize that the concatenation of Ex04 and PW embeddings (denoted as P) performed best among all phonetic feature combinations. Thus, we use it as phonetic feature in the fusion of modalities. The results shown in Table 6.7 suggest that the best performance is achieved by fusing textual and phonetic features.

We notice that phonetic features when fused with textual or visual features, improve the performance of both textual and visual unimodal classifiers consistently. This validates our hypothesis that phonetic features are an important factor in representing semantics, which leads to an improvement in Chinese sentiment analysis performance. A p-value of 0.007 in the paired t-test between with and without phonetic features suggested that the best performing improvement of integrating phonetic feature is statistically significant. The integration of multiple modalities could take advantages of information from different modalities. However, we notice that, in most of the cases, tri-modal models underperform bi-modal models. One disadvantage brought by using more modalities is the increase in the number of parameters. We conjecture that a larger set of learnable parameters leads to poor generalizability when the training sets in our experiments only consist of instances of less than 4000.

Furthermore, the information redundancy becomes more severe when combining features across different modalities. In other words, there might be a marginal effect

Table 6.8: Cross-domain evaluation. Datasets on the first column are the training sets. Datasets on the first row are the testing sets. The second column represents various baselines and our proposed method.

		Weibo	It168	Chn2000	Review-4	Review-5
Weibo	Hsentic		66.47	61.84	64.93	63.71
	charCBOW		67.55	64.08	62.09	67.78
	charSkipGram	-	65.29	59.60	53.22	49.49
	DISA(T+P)		73.68	66.55	69.16	71.01
It168	Hsentic	59.15		59.30	69.76	67.62
	charCBOW	57.54		65.05	72.25	68.13
	charSkipGram	54.54	-	64.68	68.19	64.38
	DISA(T+P)	63.75		68.36	77.00	74.07
Chn2000	Hsentic	56.36	60.67		52.03	44.77
	charCBOW	56.23	70.40		61.77	63.36
	charSkipGram	51.99	68.53	-	62.47	62.77
	DISA(T+P)	60.50	68.90		68.64	69.02
Review-4	Hsentic	58.15	73.55	59.22		80.55
	charCBOW	54.91	72.96	58.40		80.77
	charSkipGram	54.65	71.88	65.27	-	80.31
	DISA(T+P)	58.15	77.51	65.45		88.70
Review-5	Hsentic	58.44	74.73	69.08	83.15	
	charCBOW	56.73	72.47	57.06	85.77	
	charSkipGram	56.44	75.32	66.77	83.67	-
	DISA(T+P)	62.06	85.65	69.09	88.85	

of using additional modality. We will illustrate this point with examples. As aforementioned, Chinese character is made of symbols (or called radicals). Some symbols function as morphemes, while some symbols function as phonemes. For instance, the character ‘疯’ consists of two symbols, ‘疒’ and ‘风’. The pronunciation of ‘疯’ (feng1) is dominated by the symbol ‘风’ (feng1), which is the same for phonetic features. Meanwhile, ‘风’ contributes the most to the visual image of ‘疯’, the visual feature of ‘疯’ can also somehow encodes the information brought by ‘风’.

After we compare T with T+P and T+V, the performance increase induced by P is 1.40% higher than by V on average. It is apparent to conclude that phonetic feature is better at encoding semantics than visual features. The fusion of phonetic and textual embeddings achieve the best performance in four out of five cases. It indicates that the information encoded in the phonetic feature complements that of textual embedding.

6.3.4 Cross-domain Evaluation

In this section, we examine how our model performs across different domains and datasets in order to validate the generalizability of our proposed method. Particularly

Figure 6.4: The proportion of tokens in testing sets that also appear in training sets. Rows are training sets (T denotes the textual token and P denotes the phonetic token) Columns are testing sets.

for our model, we firstly pretrain the LSTM critic network on the training set. Then we fix the parameters of critic network and train the policy network on the same training set. Next, we co-train the LSTM critic network and policy network for 30 epochs. For other baselines, an LSTM network is trained using the same training set. By the end of each epoch, the development set of this training dataset and the other four datasets are tested. The epoch results are recorded. In the end, the testing results of the epoch which has the best development result are reported. The final results of the state-of-the-art methods are shown in Table 6.8.

Results show that all methods reduce their performance compared to single dataset experiments due to the internal diversity of different dataset. Even though, our method still performs better than other baselines by an average of 6.50% in accuracy. In addition to absolute performance, we compute the average performance loss for each method across different datasets between single dataset case and cross-dataset case. It shows that our method has the least performance drop, which is 14.25%. The performance drop for Hsentic, charCBOW and charSkipGram methods are 16.09%,

Table 6.9: Performance comparison between learned and random generated phonetic feature.

		Weibo	It168	Chn2000	Review-4	Review-5
Random phonetic feature (rand)		53.83	56.85	55.71	69.20	69.77
Learned phonetic feature	Ex0	66.49	84.21	77.82	81.36	83.24
	Ex04	67.28	84.69	78.18	81.88	83.38
	PO	64.28	82.30	77.09	83.97	82.71
	PW	67.80	83.73	77.45	85.37	84.18

15.69%, 17.16% respectively. We think it might be ascribed to that the proportion of shared phonetic tokens among datasets is larger than the portion of shared textual characters. Thus, phonetic features will have better transferability than textual features. Fig. 6.4 illustrates the proportion of common phonetic tokens as well as common textual tokens between each pair of datasets. The result in the figure agrees with our initial analysis.

6.3.5 Ablation Tests

We conduct ablation tests in two steps: validating phonetic features and integration of phonetic features. The first step validates the contribution of phonetic features. The second step examines which specific combination of phonetic features works best.

6.3.5.1 Validating phonetic feature

So far, we have examined the effectiveness of our model as a whole by comparing it with different baselines. In this section, we break down the proposed methods into a reinforcement learning framework and a set of features. First of all, we would like to validate if the performance gain mainly results from the reinforcement learning framework. To this end, we replace the phonetic features with random features. In particular, we generate random real-valued vectors as random phonetic feature for each character. Each dimension of the random phonetic feature vector is a float number between -1 to 1 sampled from a Gaussian distribution. Then, we use this random feature vector to represent each Chinese character and yielded the results in Table 6.9.

In the comparison between the learned phonetic feature and random phonetic feature, we can observe that the learned feature outperforms random feature with at

Figure 6.5: Performance comparison between phonetic ablation test groups. rand denotes random generated embeddings. Ex0/Ex04 represent Ex embeddings without/with tones. The same is for PO/PW. + denotes a concatenation operation.

least 13% in all datasets. This result indicates that the improvement of performance is due to the contribution of learned phonetic feature but not the training of classifiers. Phonetic feature itself is the cause and similar performance will not be achieved just by introducing random features.

We plot the results in Fig. 6.5 on the left to amplify the difference. Moreover, we find that, whether extracted from audio clips or learned from pinyin corpus, phonetic features that contain intonation (Ex04 and PW) perform better than those without intonation (EX0 and PO) in all our experiments.

This proves our initial argument that intonation plays an important role in representing Chinese sentiment. Nevertheless, we discover that the performance of various learned phonetic features is not persistent. PW prevails in three datasets while Ex04 wins in the other two datasets. As the best two phonetic features are either extracted from audio clips or learned from pinyin corpus, it is expected to take advantage of both sides. Thus, we propose the ablation test of the phonetic feature in different combination.

6.3.5.2 Integration of Phonetic Features

We combine both extracted phonetic features and learned phonetic features to form four variations. The results are shown in Table 6.10 and plotted in Fig. 6.5 on the right.

As expected, the combination of Ex04 and PW prevails in four datasets and performs close to the best in the remaining dataset. Specifically, when we compare Ex04+PW with Ex04, there is an average improvement of 1.43% across datasets. We believe the improvement was due to the semantic information provided by PW feature, as it was trained on the pinyin corpus. Contextual relation was designed to be encoded in embeddings. By merging embedding features to extracted features, the combination feature would also encode certain semantics, which we would show in the following section. Correspondingly, if we compare Ex04+PW with PW, the performance improvement was 0.80% on average.

Table 6.10: Performance comparison between different combinations of phonetic features

	Weibo	It168	Chn2000	Review-4	Review-5
Ex0	66.49	84.21	77.82	81.36	83.24
Ex04	67.28	84.69	78.18	81.88	83.38
PO	64.28	82.30	77.09	83.97	82.71
PW	67.80	83.73	77.45	85.37	84.18
Ex0+PO	65.45	81.82	77.09	83.98	83.38
Ex0+PW	67.80	82.30	78.91	84.84	84.71
Ex04+PO	67.14	80.38	77.45	83.80	84.84
Ex04+PW	68.19	85.17	79.27	84.67	85.24

This would be explainable due to Ex04 features extracted information that can only be conveyed in pronunciation. As we introduced in the start, the deep phonemic orthography has enabled Chinese pronunciation to encode meanings that were not represented in the text. The English text, in contrast, originally was designed to mimic pronunciation [36]. Due to the heterogeneity between textual and phonetic representation of the Chinese language, it is reasonable to unveil the magic behind Chinese phonetics. In summary, we have shown that both the intonation variation and deep phonemic orthography contributed to Chinese sentiment analysis task.

6.3.6 Visualization

In this section, we visualize four kinds of phonetic-related embeddings. The are Ex04, PW, Ex04+PW (P) and T+P.

As shown in Fig. 6.6(a), pinyins that have similar pronunciations (vowels) are close to each other in the embedding space. This observation matches our experimental purpose that the Ex04 feature will encode phonetic information (such as similarity) among different pronunciations. Secondly, as can be seen in Fig. 6.6(b), we visualize the embeddings of PW. Since it was learned on the phonetic corpus, certain semantics are expected to be encoded. In reality, we do find semantic closeness in the visualization. The squares are some examples we spotted. For instance, ‘Niu2’ and ‘Nai3’ are together due to ‘Niu2 Nai3 (milk)’. ‘Dian4’ and ‘Nao3’ are together due to ‘Dian4 Nao3 (computer)’. ‘Jian3’ and ‘Cha2’ are together due to ‘Jian3 Cha2 (inspection)’. Next, we visualize the combined embedding, Ex04+PW, which is also the main phonetic feature we use in our model in Fig. 6.6(c). Unsurprisingly, we observe that this

Figure 6.6: Selected t-SNE visualization of four kinds of phonetic-related embeddings. Circles cluster phonetic similarity. Squares cluster semantic similarity.

feature combines the characteristics both from Ex04 and PW, because this embedding clusters not only phonetic similarity but also semantic similarity. Finally, we visualize the fused embedding of T+P in Fig. 6.6(d). In addition to the characteristics displayed in Ex04+PW (P), the fused T+P appends with Chinese textual characters. For example, 沐Mu4 and 浴Yu4 stayed together because of semantics (bath). 桓Huan2 and 寰Huan2 stayed together because of phonetics. It can be concluded that the fused embeddings capture certain phonetic information from phonetic features and semantic information from textual embeddings. This shows us why phonetic-enrich text representation could level up the performance in sentiment analysis compared with pure textual representation.

6.4 Summary

Modern Chinese pronunciation system (pinyin) provides a new perspective in addition to the writing system in representing the Chinese language. Due to its deep phonemic orthography and intonation variations, it is expected to bring new contributions to the statistical representation of the Chinese language, especially for sentiment analysis.

In this chapter, we present an approach to learn phonetic information out of pinyin (both from audio clips and pinyin token corpus) and design a network to disambiguate intonations. With the learned phonetic information, we integrate it with textual and visual features to create new Chinese representations. Experiments on five datasets demonstrated the positive contribution of phonetic information to Chinese sentiment analysis.

Chapter 7

Summary and Future Work

In this chapter, we summarize the contribution of this thesis towards Chinese sentiment analysis. Afterward, we analyze the limitations of our proposed methods and suggest possible future work.

7.1 Summary of Proposed Method

Sentiment analysis has been a popular research topic in the community. During the recent past years, an abundance of research work has been developed for the general task which adapts to various different languages. Although they have shown positive effect in the Chinese language, we found several characteristics of the Chinese languages have not been utilized before. These characteristics will potentially bring obvious benefits to the Chinese sentiment analysis task. This thesis targets at learning and utilizing these characteristics to improve Chinese sentiment analysis. We enumerate the major contributions as below:

- In Chapter 3, we introduce the first concept-level sentiment knowledge base in the Chinese language. The knowledge base organizes lexical items and phrases into semantically-related clusters, which makes sentiment inference possible. In addition, fine-grained sentiment arousal score was assigned to each cluster. The knowledge base was constructed in an unsupervised mapping approach. The approach used synsets (ontologies) and glossaries from WordNet to link multiple language sources and perform word sense disambiguation. In comparison to the

state-of-the-art Chinese sentiment lexicons, our knowledge base achieved better performance in sentence sentiment analysis task on top of the two advantages.

- In Chapter 4, we propose to encode semantics among intra-character components (radicals) for sentiment analysis. Being a pictogram language, the Chinese character itself and its sub-components contain meanings. We present a hierarchical text representation method that considers both character-level semantics and radical-level semantics. We also specifically learned sentiment-enhanced radical representations from sentiment lexicons. The experimental result suggests that our hierarchical embeddings outperform popular word-level embeddings, as well as character-level embeddings in sentiment analysis.
- In Chapter 5, we present an aspect target sequence model (ATSM) to address aspect-based sentiment analysis task. Instead of computing the averaged word/character embeddings for aspect target, as most literature did, we argue that aspect target sequence should be specifically learned. Thus, we proposed an adaptive word embedding module to dynamically learn context-sensitive aspect target word embeddings and model aspect target sequence with an attentive-LSTM structure. In addition, we take into account the multi-granular (word, character and radical) representations of Chinese text to create an information fusion from three granularities, which further improves the performance. Experiments of ATSM on English datasets achieved state-of-the-art results. With the help of the fusion mechanism, representation advantages from each granularity combined to push the performance higher than ATSM alone in Chinese ABSA.
- In Chapter 6, we utilize phonetic information to enhance Chinese text representation. The Chinese pronunciation system offers two characteristics that distinguish it from other languages: deep phonemic orthography and intonation variations. These characteristics offer semantic cues complementary to the textual representation. To this end, we design four kinds of phonetic features for the textual character. Next, we develop a disambiguate intonation for sentiment analysis (DISA) network to learn the intonation of each textual character

given its sentence context and sentence sentiment polarity. The DISA network comprised of an actor-critic reinforcement network, where actions are the intentions and critic evaluates the sentence sentiment classification. Fused together with the textual representation, phonetic representations significantly and consistently outperformed the state of the arts in Chinese sentiment analysis task.

To sum up, we address the Chinese sentiment analysis task from four perspectives. We firstly contribute a concept-level sentiment knowledge base that would enrich the sentiment resource in the community. It is a small step taken, from the trend of bag-of-words towards bag-of-concepts for the Chinese language, because its fundamental elements replace conventional Chinese words with Chinese concepts. We then propose to extract intra-character semantics for sentiment analysis. This is the first trial in leveraging Chinese text compositionality characteristic for sentiment analysis. The effectiveness of the trial encourages wider participation in relevant research direction for the Chinese NLP. Next, we especially model the aspect target sequence in aspect-based sentiment analysis task and fuse with multi-granular representation. The work has not only unveiled the commonly ignored aspect target sequence issue but also proposed effective models to address it. The proposed models significantly elevated the state of the arts of aspect-based sentiment analysis for both the English and the Chinese language. Last but not least, we present that phonetic information of the Chinese language provides complementary representation power to existing textual embeddings. The drawn conclusion and approaches can also be extrapolated to languages with similar linguistic features, namely deep phonemic orthography, such as the Latin, the Hebrews and so forth. All the explored perspectives mentioned above bring novel approaches to addressing the Chinese sentiment analysis research and set a benchmark for related research in the future.

7.2 Limitations and Future Work

The concept-level sentiment knowledge base was based on an early version of Sentic-Net [37] where the number of lexical and phrasal items was limited. As a result, the

induced Chinese resource was also kept small in terms of volume. In addition to the volume, newer versions of SenticNet has been expanded with additional emotion tags, conceptual primitives etc. The method we used for the current version of Csentinet should also be applied to the latest sentiment resource to keep the Chinese sentiment resource updated.

In the work of hierarchical embeddings, as we only fused different embeddings at feature level: one possible improvement could be fusions at the model level, where we will integrate the classification results from different embeddings. Secondly, we would like to make a deeper analysis of Chinese radicals. In the thesis, we treat each radical in a character with equal importance which is not ideal. As radicals in the same character have different functions, graphemes act as pronunciation cues and morphemes act as semantic cues. Hsiao and Shillcock argued that semantic-phonetic compound (or phonetic compound) comprised about 81% of the 7000 frequent Chinese characters [124]. The graphemes in these characters may distract the semantics represented by morphemes. Thus, a weighted radical analysis within each character is expected to further improve performance.

In the work of phonetic-enriched text representation, even though our method only examines the Chinese language, it suggests greater potential for languages that also carry the deep phonemic orthography characteristic, such as Arabic and Hebrew. In the future, we could extend the work in the following directions: firstly, we would explore better fusion methods to combine different modalities instead of concatenations used in Chapter 5 and 6, such as tensor fusion [159] and so forth; secondly, we would like to extend the phonetic information to the word-level Chinese text representation. By doing so, we need to come up with appropriate methods to synthesize character-level pronunciation to word-level pronunciation. Thirdly, there are about 4% of the frequent 3500 Chinese characters are heteronyms (each of two or more characters wrote the same but not necessarily pronounced the same and have different meanings and origins), such as ‘行’ in ‘行(xíng)走’ and ‘银行(háng)’. They were automatically detected by the online parser in the thesis whose accuracy is not guaranteed. One

possible future improvement is to perform sense disambiguation before assigning the heteronym its correct pinyin token.

Appendix A

List of Publications

- **Haiyun Peng**, Yukun Ma, Yang Li, and Erik Cambria. “Learning multi-grained aspect target sequence for Chinese sentiment analysis.” *Knowledge-Based Systems* 148 (2018): 167-176.
- **Haiyun Peng**, Erik Cambria, and Amir Hussain. “A review of sentiment analysis research in Chinese language.” *Cognitive Computation* 9, no. 4 (2017): 423-435.
- **Haiyun Peng** and Erik Cambria. “CSenticNet: A Concept-level Resource for Sentiment Analysis in Chinese Language.” *Proceedings of International Conference on Computational Linguistics and Intelligent Text Processing (CiCLing)*, 90-104, 2017
- **Haiyun Peng**, Erik Cambria, and Xiaomei Zou. ”Radical-based hierarchical embeddings for chinese sentiment analysis at sentence level.” *Proceedings of Florida Artificial Intelligence Research Society Conference (FLAIRS)*, 347-352, 2017.
- **Haiyun Peng**, Soujanya Poria, Yang Li, and Erik Cambria. “Fusing Phonetic Features and Chinese Character Representation for Sentiment Analysis.” *Proceedings of International Conference on Computational Linguistics and Intelligent Text Processing (CiCLing)*, 2019.
- Yukun Ma, **Haiyun Peng**, and Erik Cambria. “Targeted Aspect-Based Sentiment Analysis via Embedding Commonsense Knowledge into an Attentive

LSTM” Proceedings of Thirty-Second AAAI Conference on Artificial Intelligence (AAAI), 5876-5883, 2018

- Soujanya Poria, **Haiyun Peng**, Amir Hussain, Newton Howard, and Erik Cambria. “Ensemble application of convolutional neural networks and multiple kernel learning for multimodal sentiment analysis.” *Neurocomputing* 261 (2017): 217-230.
- Yukun Ma, **Haiyun Peng**, Tahir Khan, Erik Cambria, and Amir Hussain. “Sentic LSTM: a Hybrid Network for Targeted Aspect-Based Sentiment Analysis” *Cognitive Computation* (2018): 1-12.
- Vilares David, **Haiyun Peng**, Satapathy Ranjan, and Erik Cambria. “Bable-SenticNet: A Commonsense Reasoning Framework for Multilingual Sentiment Analysis.” *Proceedings of IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2018)*.

List of Submissions

- **Haiyun Peng**, Yukun Ma, Soujanya Poria, Yang Li, and Erik Cambria. “Phonetic-enrich Text Representation for Chinese sentiment Analysis with Reinforcement Learning.” In submission to *Information Fusion*, under review.
- **Haiyun Peng**, Soujanya Poria, and Erik Cambria. “Demystify the Black Box Effect of Neural Networks: An Interpretable Approach to Chinese Sentiment Analysis.” In submission to *ACM Trans Asian and Low-Resource Language Information Processing (TALLIP)*, under review.

References

- [1] E. Cambria and B. White, “Jumping nlp curves: A review of natural language processing research,” *IEEE Computational Intelligence Magazine*, vol. 9, no. 2, pp. 48–57, 2014.
- [2] E. Cambria, D. Das, S. Bandyopadhyay, and A. Feraco, *A Practical Guide to Sentiment Analysis*. Cham, Switzerland: Springer, 2017.
- [3] S. Poria, E. Cambria, R. Bajpai, and A. Hussain, “A review of affective computing: From unimodal analysis to multimodal fusion,” *Information Fusion*, vol. 37, pp. 98–125, 2017.
- [4] S. Poria, E. Cambria, D. Hazarika, N. Mazumder, A. Zadeh, and L.-P. Morency, “Context-dependent sentiment analysis in user-generated videos,” in *Association for Computational Linguistics*, 2017, pp. 873–883.
- [5] I. Chaturvedi, E. Cambria, R. Welsch, and F. Herrera, “Distinguishing between facts and opinions for sentiment analysis: Survey and challenges,” *Information Fusion*, vol. 44, pp. 65–77, 2018.
- [6] I. Chaturvedi, E. Cambria, and D. Vilares, “Lyapunov filtering of objectivity for Spanish sentiment model,” in *International Joint Conference on Neural Networks*, Vancouver, 2016, pp. 4474–4481.
- [7] D. Rajagopal, E. Cambria, D. Olsher, and K. Kwok, “A graph-based approach to commonsense concept extraction and semantic similarity detection,” in *International World Wide Web Conference*, Rio De Janeiro, 2013, pp. 565–570.

- [8] S. Poria, E. Cambria, and A. Gelbukh, “Aspect extraction for opinion mining with a deep convolutional neural network,” *Knowledge-Based Systems*, vol. 108, pp. 42–49, 2016.
- [9] S. Poria, E. Cambria, D. Hazarika, and P. Vij, “A deeper look into sarcastic tweets using deep convolutional neural networks,” in *International Conference on Computational Linguistics*, 2016, pp. 1601–1612.
- [10] Y. Ma, E. Cambria, and S. Gao, “Label embedding for zero-shot fine-grained named entity typing,” in *International Conference on Computational Linguistics*, 2016, pp. 171–180.
- [11] N. Majumder, S. Poria, A. Gelbukh, and E. Cambria, “Deep learning-based document modeling for personality detection from text,” *IEEE Intelligent Systems*, vol. 32, no. 2, pp. 74–79, 2017.
- [12] S. Poria, I. Chaturvedi, E. Cambria, and A. Hussain, “Convolutional MKL based multimodal emotion recognition and sentiment analysis,” in *IEEE International Conference on Data Mining*, Barcelona, 2016, pp. 439–448.
- [13] R. Mihalcea and A. Garimella, “What men say, what women hear: Finding gender-specific meaning shades,” *IEEE Intelligent Systems*, vol. 31, no. 4, pp. 62–67, 2016.
- [14] Y. Li, Q. Pan, T. Yang, S. Wang, J. Tang, and E. Cambria, “Learning word representations for sentiment analysis,” *Cognitive Computation*, vol. 9, no. 6, pp. 843–851, 2017.
- [15] A. Hussain and E. Cambria, “Semi-supervised learning for big social data analysis,” *Neurocomputing*, vol. 275, pp. 1662–1673, 2018.
- [16] L. Oneto, F. Bisio, E. Cambria, and D. Anguita, “Statistical learning theory and ELM for big social data analysis,” *IEEE Computational Intelligence Magazine*, vol. 11, no. 3, pp. 45–55, 2016.

- [17] A. Bandhakavi, N. Wiratunga, S. Massie, and P. Deepak, “Lexicon generation for emotion analysis of text,” *IEEE Intelligent Systems*, vol. 32, no. 1, pp. 102–108, 2017.
- [18] M. Dragoni, S. Poria, and E. Cambria, “OntoSenticNet: A commonsense ontology for sentiment analysis,” *IEEE Intelligent Systems*, vol. 33, no. 3, pp. 77–85, 2018.
- [19] E. Cambria, S. Poria, D. Hazarika, and K. Kwok, “SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings,” in *Association for the Advancement of Artificial Intelligence*, 2018, pp. 1795–1802.
- [20] E. Cambria and A. Hussain, *Sentic Computing: A Common-Sense-Based Framework for Concept-Level Sentiment Analysis*. Cham, Switzerland: Springer, 2015.
- [21] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, “Semi-supervised recursive autoencoders for predicting sentiment distributions,” in *Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 151–161.
- [22] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, C. Potts *et al.*, “Recursive deep models for semantic compositionality over a sentiment treebank,” in *Empirical Methods in Natural Language Processing*, vol. 1631. Citeseer, 2013, p. 1642.
- [23] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, and K. Xu, “Adaptive recursive neural network for target-dependent twitter sentiment classification.” in *Association for Computational Linguistics (2)*, 2014, pp. 49–54.
- [24] Q. Qian, B. Tian, M. Huang, Y. Liu, X. Zhu, and X. Zhu, “Learning tag embeddings and tag-specific composition functions in recursive neural network.” in *Association for Computational Linguistics (1)*, 2015, pp. 1365–1374.

- [25] Y. Kim, “Convolutional neural networks for sentence classification,” *arXiv preprint arXiv:1408.5882*, 2014.
- [26] C. N. Dos Santos and M. Gatti, “Deep convolutional neural networks for sentiment analysis of short texts.” in *International Conference on Computational Linguistics*, 2014, pp. 69–78.
- [27] S. Poria, H. Peng, A. Hussain, N. Howard, and E. Cambria, “Ensemble application of convolutional neural networks and multiple kernel learning for multi-modal sentiment analysis,” *Neurocomputing*, 2017.
- [28] S. Sukhbaatar, J. Weston, R. Fergus *et al.*, “End-to-end memory networks,” in *Advances in neural information processing systems*, 2015, pp. 2440–2448.
- [29] A. M. Rush, S. Chopra, and J. Weston, “A neural attention model for abstractive sentence summarization,” *arXiv preprint arXiv:1509.00685*, 2015.
- [30] D. Tang, B. Qin, and T. Liu, “Aspect level sentiment classification with deep memory network,” in *Empirical Methods in Natural Language Processing*, 2016, pp. 214–224.
- [31] A. Graves, G. Wayne, and I. Danihelka, “Neural turing machines,” *arXiv preprint arXiv:1410.5401*, 2014.
- [32] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.
- [33] Y. Wang, M. Huang, L. Zhao, and X. Zhu, “Attention-based lstm for aspect-level sentiment classification,” in *Empirical Methods in Natural Language Processing*, 2016, pp. 606–615.
- [34] R. Frost, L. Katz, and S. Bentin, “Strategies for visual word recognition and orthographical depth: a multilingual comparison.” *Journal of Experimental Psychology: Human Perception and Performance*, vol. 13, no. 1, p. 104, 1987.

- [35] L. Katz and R. Frost, “The reading process is different for different orthographies: The orthographic depth hypothesis,” *ADVANCES IN PSYCHOLOGY-AMSTERDAM*-, vol. 94, pp. 67–67, 1992.
- [36] K. H. Albrow, *The English writing system: Notes towards a description*. Longman, 1972.
- [37] E. Cambria, S. Poria, R. Bajpai, and B. Schuller, “SenticNet 4: A semantic resource for sentiment analysis based on conceptual primitives,” in *International Conference on Computational Linguistics*, 2016, pp. 2666–2677.
- [38] S. Baccianella, A. Esuli, and F. Sebastiani, “Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining.” in *Language Resources and Evaluation Conference*, vol. 10, 2010, pp. 2200–2204.
- [39] Z. Dong and Q. Dong, *HowNet and the Computation of Meaning*. World Scientific, 2006.
- [40] L.-W. Ku, Y.-T. Liang, and H.-H. Chen, “Opinion extraction, summarization and tracking in news and blog corpora.” in *Association for the Advancement of Artificial Intelligence spring symposium: Computational approaches to analyzing weblogs*, 2006.
- [41] X. Chen, L. Xu, Z. Liu, M. Sun, and H.-B. Luan, “Joint learning of character and word embeddings.” in *International Joint Conferences on Artificial Intelligence*, 2015, pp. 1236–1242.
- [42] Y. Sun, L. Lin, N. Yang, Z. Ji, and X. Wang, “Radical-enhanced chinese character embedding,” in *Lecture Notes in Computer Science*, vol. 8835, 2014, pp. 279–286.
- [43] X. Shi, J. Zhai, X. Yang, Z. Xie, and C. Liu, “Radical embedding: Delving deeper to chinese radicals.” in *Association for Computational Linguistics (2)*, 2015, pp. 594–598.

- [44] R. Yin, Q. Wang, R. Li, P. Li, and B. Wang, “Multi-granularity chinese word embedding,” in *Empirical Methods in Natural Language Processing*, 2016, pp. 981–986.
- [45] D. Tang, B. Qin, X. Feng, and T. Liu, “Effective lstms for target-dependent sentiment classification,” *arXiv preprint arXiv:1512.01100*, 2015.
- [46] D. Tang, B. Qin, and T. Liu, “Document modeling with gated recurrent neural network for sentiment classification,” in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 1422–1432.
- [47] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. 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, pp. 1480–1489.
- [48] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up?: sentiment classification using machine learning techniques,” in *Proceedings of Empirical methods in natural language processing- Volume 10*. Association for Computational Linguistics, 2002, pp. 79–86.
- [49] D. Ma, S. Li, X. Zhang, and H. Wang, “Interactive attention networks for aspect-level sentiment classification,” in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. Association for the Advancement of Artificial Intelligence Press, 2017, pp. 4068–4074.
- [50] H. Peng, Y. Ma, Y. Li, and E. Cambria, “Learning multi-grained aspect target sequence for chinese sentiment analysis,” *Knowledge-Based Systems*, vol. 148, pp. 167–176, 2018.
- [51] Y. Li, W. Li, F. Sun, and S. Li, “Component-enhanced chinese character embeddings,” *arXiv preprint arXiv:1508.06669*, 2015.

- [52] T.-r. Su and H.-y. Lee, “Learning chinese word representations from glyphs of characters,” in *Empirical Methods in Natural Language Processing*, 2017, pp. 264–273.
- [53] S. Cao, W. Lu, J. Zhou, and X. Li, “cw2vec: Learning chinese word embeddings with stroke n-gram information,” in *Association for the Advancement of Artificial Intelligence*, 2018.
- [54] C. Quan and F. Ren, “Construction of a blog emotion corpus for chinese emotional expression analysis,” in *Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2009, pp. 1446–1454.
- [55] Y. Zhao, B. Qin, and T. Liu, “Creating a fine-grained corpus for chinese sentiment analysis,” *IEEE Intelligent Systems*, vol. 30, no. 5, pp. 36–43, 2014.
- [56] H.-H. Wu, A. C.-R. Tsai, R. T.-H. Tsai, and J. Y.-j. Hsu, “Building a graded chinese sentiment dictionary based on commonsense knowledge for sentiment analysis of song lyrics,” *Journal of Information Science and Engineering*, vol. 29, no. 4, pp. 647–662, 2013.
- [57] Y. Su and S. Li, “Constructing chinese sentiment lexicon using bilingual information,” in *Chinese Lexical Semantics*. Springer, 2013, pp. 322–331.
- [58] L. Liu, M. Lei, and H. Wang, “Combining domain-specific sentiment lexicon with hownet for chinese sentiment analysis,” *Journal of Computers*, vol. 8, no. 4, pp. 878–883, 2013.
- [59] H. Xu, K. Zhao, L. Qiu, and C. Hu, “Expanding chinese sentiment dictionaries from large scale unlabeled corpus.” in *Pacific Asia Conference on Language, Information and Computation*, 2010, pp. 301–310.
- [60] G. Xu, X. Meng, and H. Wang, “Build chinese emotion lexicons using a graph-based algorithm and multiple resources,” in *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 1209–1217.

- [61] B. Wang, Y. Huang, X. Wu, and X. Li, “A fuzzy computing model for identifying polarity of chinese sentiment words,” *Computational Intelligence and Neuroscience*, vol. 2015, 2015.
- [62] S. Tan and J. Zhang, “An empirical study of sentiment analysis for chinese documents,” *Expert Systems with Applications*, vol. 34, no. 4, pp. 2622–2629, 2008.
- [63] Z. Zhai, H. Xu, B. Kang, and P. Jia, “Exploiting effective features for chinese sentiment classification,” *Expert Systems with Applications*, vol. 38, no. 8, pp. 9139–9146, 2011.
- [64] Z. Su, H. Xu, D. Zhang, and Y. Xu, “Chinese sentiment classification using a neural network tool?word2vec,” in *Multisensor Fusion and Information Integration for Intelligent Systems (MFI), 2014 International Conference on*. IEEE, 2014, pp. 1–6.
- [65] L. Xiang, “Ideogram based chinese sentiment word orientation computation,” *arXiv preprint arXiv:1110.4248*, 2011.
- [66] W. Xu, Z. Liu, T. Wang, and S. Liu, “Sentiment recognition of online chinese micro movie reviews using multiple probabilistic reasoning model,” *Journal of Computers*, vol. 8, no. 8, pp. 1906–1911, 2013.
- [67] Y. Cao, Z. Chen, R. Xu, T. Chen, and L. Gui, “A joint model for chinese microblog sentiment analysis,” *Association for Computational Linguistics-International Joint Conference on Natural Language Processing 2015*, p. 61, 2015.
- [68] L. Liu, D. Luo, M. Liu, J. Zhong, Y. Wei, and L. Sun, “A self-adaptive hidden markov model for emotion classification in chinese microblogs,” *Mathematical Problems in Engineering*, 2015.

- [69] L.-W. Ku, T.-H. Huang, and H.-H. Chen, “Using morphological and syntactic structures for chinese opinion analysis,” in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*. Association for Computational Linguistics, 2009, pp. 1260–1269.
- [70] W. Xiong, Y. Jin, and Z. Liu, “Chinese sentiment analysis using appraiser-degree-negation combinations and pso,” *Journal of Computers*, vol. 9, no. 6, pp. 1410–1417, 2014.
- [71] H.-P. Zhang, H.-K. Yu, D.-Y. Xiong, and Q. Liu, “Hhmm-based chinese lexical analyzer ictclas,” in *Proceedings of the second SIGHAN workshop on Chinese language processing-Volume 17*. Association for Computational Linguistics, 2003, pp. 184–187.
- [72] C. Zhang, D. Zeng, J. Li, F.-Y. Wang, and W. Zuo, “Sentiment analysis of chinese documents: From sentence to document level,” *Journal of the American Society for Information Science and Technology*, vol. 60, no. 12, pp. 2474–2487, 2009.
- [73] T. Zagibalov and J. Carroll, “Automatic seed word selection for unsupervised sentiment classification of chinese text,” in *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2008, pp. 1073–1080.
- [74] T. Z. J. Carroll, “Unsupervised classification of sentiment and objectivity in chinese text,” in *Third International Joint Conference on Natural Language Processing*, 2008, p. 304.
- [75] R. Li, S. Shi, H. Huang, C. Su, and T. Wang, “A method of polarity computation of chinese sentiment words based on gaussian distribution,” in *Computational Linguistics and Intelligent Text Processing*. Springer, 2014, pp. 53–61.
- [76] S. Zhuo, X. Wu, and X. Luo, “Chinese text sentiment analysis based on fuzzy semantic model,” in *Cognitive Informatics & Cognitive Computing (ICCI* CC), 2014 IEEE 13th International Conference on*. IEEE, 2014, pp. 535–540.

- [77] Q. Su, X. Xu, H. Guo, Z. Guo, X. Wu, X. Zhang, B. Swen, and Z. Su, “Hidden sentiment association in chinese web opinion mining,” in *Proceedings of the 17th international conference on World Wide Web*. ACM, 2008, pp. 959–968.
- [78] S.-J. Wu and R.-D. Chiang, “Using syntactic rules to combine opinion elements in chinese opinion mining systems,” *Journal of Convergence Information Technology*, vol. 10, no. 2, p. 137, 2015.
- [79] P. Zhang and Z. He, “A weakly supervised approach to chinese sentiment classification using partitioned self-training,” *Journal of Information Science*, vol. 39, no. 6, pp. 815–831, 2013.
- [80] B. Yuan, Y. Liu, H. Li, T. T. T. PHAN, G. Kausar, C. N. Sing-Bik, and W. Wahi, “Sentiment classification in chinese microblogs: Lexicon-based and learning-based approaches,” *International Proceedings of Economics Development and Research (IPEDR)*, vol. 68, 2013.
- [81] X. Wan, “Using bilingual knowledge and ensemble techniques for unsupervised chinese sentiment analysis,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2008, pp. 553–561.
- [82] Y. He, H. Alani, and D. Zhou, “Exploring english lexicon knowledge for chinese sentiment analysis,” in *CIPS-SIGHAN Joint conference on Chinese language processing*, 2010.
- [83] T. McArthur and F. McArthur, *The Oxford companion to the English language*, ser. Oxford Companions Series. Oxford University Press, 1992.
- [84] C. Li, B. Xu, G. Wu, S. He, G. Tian, and H. Hao, “Recursive deep learning for sentiment analysis over social data,” in *IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*. IEEE Computer Society, 2014, pp. 180–185.

- [85] M. Hu and B. Liu, “Mining and summarizing customer reviews,” in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 168–177.
- [86] L. Zhuang, F. Jing, and X.-Y. Zhu, “Movie review mining and summarization,” in *Proceedings of the 15th ACM international conference on Information and knowledge management*. ACM, 2006, pp. 43–50.
- [87] C. Toprak, N. Jakob, and I. Gurevych, “Sentence and expression level annotation of opinions in user-generated discourse,” in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 575–584.
- [88] S. Y. M. Lee and Z. Wang, “Emotion in code-switching texts: Corpus construction and analysis,” *Association for Computational Linguistics-International Joint Conference on Natural Language Processing 2015*, p. 91, 2015.
- [89] B. Liu, “Sentiment analysis and opinion mining,” *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1–167, 2012.
- [90] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell, “Toward an architecture for never-ending language learning.” in *Association for the Advancement of Artificial Intelligence*, vol. 5, 2010, p. 3.
- [91] Z. Dong, Q. Dong, and C. Hao, “Hownet and its computation of meaning,” in *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*. Association for Computational Linguistics, 2010, pp. 53–56.
- [92] D. Gao, F. Wei, W. Li, X. Liu, and M. Zhou, “Cross-lingual sentiment lexicon learning with bilingual word graph label propagation,” *Computational Linguistics*, 2015.
- [93] Z. Li and M. Sun, “Punctuation as implicit annotations for chinese word segmentation,” *Computational Linguistics*, vol. 35, no. 4, pp. 505–512, 2009.

- [94] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [95] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. ” O’Reilly Media, Inc.”, 2009.
- [96] P. Eckman, “Universal and cultural differences in facial expression of emotion,” in *Nebraska symposium on motivation*, vol. 19. University of Nebraska Press Lincoln, 1972, pp. 207–284.
- [97] G. Wei, H. An, T. Dong, and H. Li, “A novel micro-blog sentiment analysis approach by longest common sequence and k-medoids,” *Pacific Asia Conference on Information Systems*, p. 38, 2014.
- [98] S. M. Liu and J.-H. Chen, “A multi-label classification based approach for sentiment classification,” *Expert Systems with Applications*, vol. 42, no. 3, pp. 1083–1093, 2015.
- [99] C. Quan, X. Wei, and F. Ren, “Combine sentiment lexicon and dependency parsing for sentiment classification,” in *System Integration (SII), 2013 IEEE/SICE International Symposium on*. IEEE, 2013, pp. 100–104.
- [100] Q. Li, Q. Zhi, and M. Li, “An combined sentiment classification system for sishan-8,” *Association for Computational Linguistics-International Joint Conference on Natural Language Processing 2015*, p. 74, 2015.
- [101] S. Wen and X. Wan, “Emotion classification in microblog texts using class sequential rules,” in *Association for the Advancement of Artificial Intelligence*, 2014.

- [102] B. Wei and C. Pal, “Cross lingual adaptation: an experiment on sentiment classifications,” in *Proceedings of the Association for Computational Linguistics 2010 Conference Short Papers*. Association for Computational Linguistics, 2010, pp. 258–262.
- [103] S. Poria, I. Chaturvedi, E. Cambria, and F. Bisio, “Sentic LDA: Improving on LDA with semantic similarity for aspect-based sentiment analysis,” in *International Joint Conference on Neural Networks*, 2016, pp. 4465–4473.
- [104] B. Lu, C. Tan, C. Cardie, and B. K. Tsou, “Joint bilingual sentiment classification with unlabeled parallel corpora,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 320–330.
- [105] H. Zhou, L. Chen, F. Shi, and D. Huang, “Learning bilingual sentiment word embeddings for cross-language sentiment classification,” in *Association for Computational Linguistics*, 2015.
- [106] Q. Chen, W. Li, Y. Lei, X. Liu, and Y. He, “Learning to adapt credible knowledge in cross-lingual sentiment analysis,” in *Association for Computational Linguistics*, 2015.
- [107] Y. Zhang, M. M. Rahman, A. Braylan, B. Dang, H.-L. Chang, H. Kim, Q. McNamara, A. Angert, E. Banner, V. Khetan *et al.*, “Neural information retrieval: A literature review,” *arXiv preprint arXiv:1611.06792*, 2016.
- [108] J. Turian, L. Ratinov, and Y. Bengio, “Word representations: a simple and general method for semi-supervised learning,” in *Association for Computational Linguistics*, 2010, pp. 384–394.
- [109] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A neural probabilistic language model,” *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

- [110] A. Mnih and G. Hinton, “Three new graphical models for statistical language modelling,” in *International Conference on Machine Learning*, 2007, pp. 641–648.
- [111] A. Mnih and G. E. Hinton, “A scalable hierarchical distributed language model,” in *Neural Information Processing Systems*, 2009, pp. 1081–1088.
- [112] A. Mnih and K. Kavukcuoglu, “Learning word embeddings efficiently with noise-contrastive estimation,” in *Neural Information Processing Systems*, 2013, pp. 2265–2273.
- [113] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, “Recurrent neural network based language model.” in *Interspeech*, vol. 2, 2010, p. 3.
- [114] R. Collobert and J. Weston, “A unified architecture for natural language processing: Deep neural networks with multitask learning,” in *International Conference on Machine Learning*. ACM, 2008, pp. 160–167.
- [115] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv preprint arXiv:1301.3781*, 2013.
- [116] W. Y. Zou, R. Socher, D. M. Cer, and C. D. Manning, “Bilingual word embeddings for phrase-based machine translation.” in *Empirical Methods in Natural Language Processing*, 2013, pp. 1393–1398.
- [117] X. Zheng, H. Chen, and T. Xu, “Deep learning for chinese word segmentation and pos tagging.” in *Empirical Methods in Natural Language Processing*, 2013, pp. 647–657.
- [118] M. Sun, X. Chen, K. Zhang, Z. Guo, and Z. Liu, “Thulac: An efficient lexical analyzer for chinese,” Technical Report, Tech. Rep., 2016.
- [119] J. Xu, J. Liu, L. Zhang, Z. Li, and H. Chen, “Improve chinese word embeddings by exploiting internal structure,” in *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1041–1050.

- [120] Y. Li, W. Li, F. Sun, and S. Li, “Component-enhanced chinese character embeddings,” in *Empirical Methods in Natural Language Processing*, 2015, pp. 829–834.
- [121] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in neural information processing systems*, 2013, pp. 3111–3119.
- [122] F. Liu, H. Lu, C. Lo, and G. Neubig, “Learning character-level compositionality with visual features,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2017, pp. 2059–2068.
- [123] H. Shu, R. C. Anderson, and N. Wu, “Phonetic awareness: Knowledge of orthography–phonology relationships in the character acquisition of chinese children.” *Journal of Educational Psychology*, vol. 92, no. 1, p. 56, 2000.
- [124] J. H.-w. Hsiao and R. Shillcock, “Analysis of a chinese phonetic compound database: Implications for orthographic processing,” *Journal of psycholinguistic research*, vol. 35, no. 5, pp. 405–426, 2006.
- [125] R. Mihalcea, C. Banea, and J. Wiebe, “Learning multilingual subjective language via cross-lingual projections,” in *Proceedings of the 45th annual meeting of the association of computational linguistics*, 2007, pp. 976–983.
- [126] L. Gui, R. Xu, Q. Lu, J. Xu, J. Xu, B. Liu, and X. Wang, “Cross-lingual opinion analysis via negative transfer detection.” in *Association for Computational Linguistics (2)*, 2014, pp. 860–865.
- [127] S. Jain and S. Batra, “Cross-lingual sentiment analysis using modified brae,” in *Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2015, pp. 159–168.

- [128] P. Lambert, “Aspect-level cross-lingual sentiment classification with constrained smt,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers)*. Association for Computational Linguistics, 2015, pp. 781–787.
- [129] C. Fellbaum, *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.
- [130] F. Bond and R. Foster, “Linking and extending an open multilingual wordnet,” in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2013, pp. 1352–1362.
- [131] M. Lesk, “Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone,” in *Proceedings of the 5th annual international conference on Systems documentation*. ACM, 1986, pp. 24–26.
- [132] T. Baldwin, S. Kim, F. Bond, S. Fujita, D. Martinez, and T. Tanaka, “A re-examination of mrd-based word sense disambiguation,” *ACM Transactions on Asian Language Information Processing (TALIP)*, vol. 9, no. 1, p. 4, 2010.
- [133] A. Pavlenko, “Emotions and the body in russian and english,” *Pragmatics & Cognition*, vol. 10, no. 1, pp. 207–241, 2002.
- [134] A. Wierzbicka, “Preface: Bilingual lives, bilingual experience,” *Journal of multilingual and multicultural development*, vol. 25, no. 2-3, pp. 94–104, 2004.
- [135] H. Peng, E. Cambria, and A. Hussain, “A review of sentiment analysis research in chinese language,” *Cognitive Computation*, 2017.
- [136] X. Rong, “word2vec parameter learning explained,” *arXiv preprint arXiv:1411.2738*, 2014.
- [137] G. Qiu, B. Liu, J. Bu, and C. Chen, “Opinion word expansion and target extraction through double propagation,” *Computational linguistics*, vol. 37, no. 1, pp. 9–27, 2011.

- [138] J. Yu, Z.-J. Zha, M. Wang, and T.-S. Chua, “Aspect ranking: identifying important product aspects from online consumer reviews,” in *49th Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 1496–1505.
- [139] W. Wang, S. J. Pan, D. Dahlmeier, and X. Xiao, “Coupled multi-layer attentions for co-extraction of aspect and opinion terms.” in *Association for the Advancement of Artificial Intelligence*, 2017, pp. 3316–3322.
- [140] K. Schouten and F. Frasincar, “Survey on aspect-level sentiment analysis,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 3, pp. 813–830, 2016.
- [141] J. Zhu, H. Wang, B. K. Tsou, and M. Zhu, “Multi-aspect opinion polling from textual reviews,” in *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 2009, pp. 1799–1802.
- [142] T. Mullen and N. Collier, “Sentiment analysis using support vector machines with diverse information sources.” in *Empirical Methods in Natural Language Processing*, vol. 4, 2004, pp. 412–418.
- [143] S. Kiritchenko, X. Zhu, C. Cherry, and S. Mohammad, “Nrc-canada-2014: Detecting aspects and sentiment in customer reviews,” in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 2014, pp. 437–442.
- [144] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [145] H. Peng, E. Cambria, and X. Zou, “Radical-based hierarchical embeddings for chinese sentiment analysis at sentence level,” in *Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference*, 2017, pp. 347–352.

- [146] W. Che, Y. Zhao, H. Guo, Z. Su, and T. Liu, “Sentence compression for aspect-based sentiment analysis,” *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 23, no. 12, pp. 2111–2124, 2015.
- [147] Y. Zhao, B. Qin, and T. Liu, “Creating a fine-grained corpus for chinese sentiment analysis,” *IEEE Intelligent Systems*, vol. 30, no. 1, pp. 36–43, 2015.
- [148] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, “Semeval-2014 task 4: Aspect based sentiment analysis,” *Proceedings of SemEval*, pp. 27–35, 2014.
- [149] C. Huang and H. Zhao, “Chinese word segmentation: A decade review,” *Journal of Chinese Information Processing*, vol. 21, no. 3, pp. 8–20, 2007.
- [150] R. Yin, Q. Wang, P. Li, R. Li, and B. Wang, “Multi-granularity chinese word embedding,” in *Empirical Methods in Natural Language Processing*, 2016, pp. 981–986.
- [151] C. Hansen, “Chinese ideographs and western ideas,” *The Journal of Asian Studies*, vol. 52, no. 2, pp. 373–399, 1993.
- [152] T. Zhang, M. Huang, and L. Zhao, “Learning structured representation for text classification via reinforcement learning,” in *Association for the Advancement of Artificial Intelligence*. Association for the Advancement of Artificial Intelligence, 2018.
- [153] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Empirical Methods in Natural Language Processing*, 2014, pp. 1532–1543.
- [154] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, “Stacked convolutional auto-encoders for hierarchical feature extraction,” in *International Conference on Artificial Neural Networks*. Springer, 2011, pp. 52–59.

- [155] A. Benjamin, “History and prospect of chinese romanization,” *Chinese Librarianship*, 1997.
- [156] F. Eyben, M. Wöllmer, and B. Schuller, “Opensmile: the munich versatile and fast open-source audio feature extractor,” in *Proceedings of the 18th ACM international conference on Multimedia*. ACM, 2010, pp. 1459–1462.
- [157] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, “Policy gradient methods for reinforcement learning with function approximation,” in *Advances in neural information processing systems*, 2000, pp. 1057–1063.
- [158] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.
- [159] A. Zadeh, M. Chen, S. Poria, E. Cambria, and L.-P. Morency, “Tensor fusion network for multimodal sentiment analysis,” in *Empirical Methods in Natural Language Processing*, 2017, pp. 1103–1114.
- [160] C. G. Snoek, M. Worring, and A. W. Smeulders, “Early versus late fusion in semantic video analysis,” in *Proceedings of the 13th annual ACM international conference on Multimedia*. ACM, 2005, pp. 399–402.
- [161] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.
- [162] C. yu Tseng, *An acoustic phonetic study on tones in Mandarin Chinese*. Institute of History & Philosophy, 1990, vol. 94.