12-4-2017

Aspect Based Sentiment Analysis On Review Data

Wei Xue
wxue004@cs.fiu.edu

DOI: 10.25148/etd.FIDC004076

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

ASPECT BASED SENTIMENT ANALYSIS ON REVIEW DATA

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Wei Xue

2018

To: Dean John L. Volakis
College of Engineering & Computing

This dissertation, written by Wei Xue, and entitled Aspect Based Sentiment Analysis on Review Data, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

Sundaraja Sitharama Iyengar

Jaime Leonardo Bobadilla

Ning Xie

Debra VanderMeer

Tao Li, Major Professor

Date of Defense: Dec 4, 2017

The dissertation of Wei Xue is approved.

Dean John L. Volakis
College of Engineering & Computing

Andres G. Gil
Vice President for Research and Economic Development
and Dean of the University Graduate School

Florida International University, 2018

DEDICATION

To my parents.

ACKNOWLEDGMENTS

First and foremost, I would like to express my gratitude to my advisor Dr. Tao Li for his support and guidance. Without his help, I would not have enjoyable research life at FIU, and this dissertation would not have existed. Second, I would like to extend my thanks to all my dissertation committee members: Dr. S.S. Iyengar, Dr. Jaime Leonardo Bobadilla, Dr. Ning Xie, and Dr. Debra VanderMeer for their helpful advices on my research and career plans. Third, I would like to thank all other my coauthors and labmates. It is a great honor for me to join Knowledge Discovery and Research Group (KDRG) and work with them. I am very grateful to all my colleagues of KDRG including Dr. Jingxuan Li, Dr. Li Zheng, Dr. Liang Tang, Dr. Lei Li, Dr. Chao Shen, Dr. Yexi Jiang, Dr. Longhui Zhang, Dr. Chunqiu Zeng, Dr. Wubai Zhou, Hongtai Li, Xiaolong Zhu, Wentao Wang, Boyuan Guan, Qing Wang, Ramesh Baral and Shekoofeh Mokhtari. Finally, I would like to thank my parents and family for their love and understanding.

ABSTRACT OF THE DISSERTATION

ASPECT BASED SENTIMENT ANALYSIS ON REVIEW DATA

by

Wei Xue

Florida International University, 2018

Miami, Florida

Professor Tao Li, Major Professor

With proliferation of user-generated reviews, new opportunities and challenges arise. The advance of Web technologies allows people to access a large amount of reviews of products and services online. Knowing what others like and dislike becomes increasingly important for their decision making in online shopping. The retailers also care more than ever about online reviews, because a vast pool of reviews enables them to monitor reputations and collect feedbacks efficiently. However, people often find difficult times in identifying and summarizing fine-grained sentiments buried in the opinion-rich resources. The traditional sentiment analysis, which focuses on the overall sentiments, fails to uncover the sentiments with regard to the aspects of the reviewed entities.

This dissertation studied the research problem of Aspect Based Sentiment Analysis (ABSA), which is to reveal the aspect-dependent sentiment information of review text. ABSA consists of several subtasks: 1) aspect extraction, 2) aspect term extraction, 3) aspect category classification, and 4) sentiment polarity classification at aspect level. We focused on the approach of topic models and neural networks for ABSA. First, to extract the aspects from a collection of reviews and to detect the sentiment polarity regarding the aspects in each review, we proposed a few probabilistic graphical models, which can model words distribution in reviews and aspect ratings at the same time. Second, we presented a multi-task learning model based on long-short term memory and convolutional neural network for aspect category classification and aspect term extraction. Third, for aspect-level sentiment polarity classification, we developed a gated convolution neural network, which can be applied to aspect category sentiment analysis as well as aspect target sentiment analysis.

TABLE OF CONTENTS

CHAPTER PAGE

1. Introduction 1
1.1 Background 1
1.2 Aspect Based Sentiment Analysis 2
1.3 Problem Definitions 4
1.4 Summary and Roadmap 6

2. Topic Modeling: Aspect Extraction and Rating Inference for Hotel Reviews . . . 7
2.1 Introduction 7
2.2 Related Work 9
2.3 Problem Formulation 10
2.4 Proposed Models 11
2.4.1 Assumptions 12
2.4.2 Motivation 13
2.4.3 ARID Model 14
2.4.4 ARIH Model 15
2.4.5 Estimation 17
2.4.6 Incorporating Prior Knowledge 20
2.4.7 Prediction 20
2.4.8 Experiments 21
2.5 Conclusion 26

3. Deep Learning Models for Aspect Based Sentiment Analysis 27
3.1 Introduction 27
3.2 Basic Modules in Deep Learning 32
3.2.1 Word Embeddings 32
3.2.2 CNN 34
3.2.3 LSTM 37
3.2.4 Memory Networks and Attention Mechanism 38
3.3 Training Strategy 40
3.4 Advances in Deep Learning Models on Reviews 42
3.5 Polarity Classification 42
3.5.1 Neural Networks for Text Classification 44
3.5.2 Neural Networks for Sentiment Polarity Classification 57
3.6 Extraction and Polarity Classification for Aspect Terms 58
3.7 Aspect Category Polarity Classification 61
3.8 Review Summarization and Generation 62
3.9 Conclusion 63

4. MTNA: A Neural Multi-task Model for Aspect Category Classification and Aspect Term Extraction On Restaurant Reviews . . . 65
4.1 Introduction . . . 65
4.2 Related Work . . . 67
4.3 Problem Definition . . . 69
4.4 The Multi-task Learning Model . . . 69
4.5 Experiments . . . 73
4.5.1 Datasets . . . 73
4.5.2 Experiment Setup . . . 74
4.5.3 Compared Methods . . . 75
4.5.4 Results and Analysis . . . 76
4.6 Conclusion . . . 78

5. Gated Convolutional Neural Networks for Aspect Based Sentiment Analysis . . . 79
5.1 Introduction . . . 79
5.2 Related Work . . . 81
5.2.1 Neural Networks . . . 81
5.2.2 Aspect based Sentiment Analysis . . . 82
5.3 The Formal Definition of ACSA and ATSA . . . 84
5.4 Gated Convolutional Network with Aspect Embedding on ACSA . . . 84
5.5 Gating Mechanisms . . . 87
5.6 Gated Convolutional Network with Aspect Embedding on ATSA . . . 88
5.7 Experiments . . . 89
5.7.1 Datasets and Experiment Preparation . . . 89
5.7.2 Compared Methods . . . 91
5.7.3 Results and Analysis . . . 92
5.7.4 Effects of Kernel Sizes . . . 94
5.7.5 Training Time . . . 94
5.7.6 Gating Mechanisms . . . 95
5.7.7 ATSA . . . 96
5.8 Conclusions . . . 97

6. Conclusion and Future Work . . . 98

BIBLIOGRAPHY . . . 100

VITA . . . 119

LIST OF TABLES

TABLE PAGE

1.1 A restaurant review 5

2.1 The table of notations 12

2.2 The largest variance ratio of principle component analysis 17

2.3 Seed words 22

2.4 Frequentest head terms and modifier terms 22

2.5 Performance of Aspect Inference 24

3.1 The neural networks for text classification 44

3.2 The neural networks for sentiment classification 57

4.1 The statistics of SemEval-2016 Restaurant Review Dataset 75

4.2 Comparison results in F1 scores on three datasets. 76

4.3 F1 scores of models on SE16 across six aspects 77

5.1 Two example sentences in one test dataset **M** of restaurant review dataset of SemEval 2014. 89

5.2 The statistics of the unrolled dataset for ACSA task 90

5.3 The statistics of the unrolled dataset in SemEval 2014 for ATSA task. 90

5.4 The accuracy of all models on test datasets 92

5.5 The accuracy of GCAE with convolutional kernels of different window sizes on ACSA task. 94

5.6 The model training time in seconds on ACSA task. 94

5.7 The model training time in seconds on ATSA task. 95

5.8 The accuracy of different gating units on restaurant reviews on ACSA task . . . 95

5.9 The accuracy of ATSA subtask on SemEval 2014 Task 4 96

LIST OF FIGURES

FIGURE PAGE

1.1 A hotel review example 3

2.1 Graphical Representation of ARID model. The outer box represents D reviews, while the inner box contains M phrases 14

2.2 Graphical Representation of ARIH model 16

2.3 Aspect Rating Dispersity of Hotels 25

2.4 Aspect Rating Prediction on Reviews of Different Dispersities 25

3.1 The diagram of skip-gram 33

3.2 The diagram of CNN for text classification 35

4.1 MTNA on a sequence of five words 72

5.1 Illustration of our model GCAE for ACSA task 86

5.2 Illustration of model GCAE for ATSA task 88

CHAPTER 1

Introduction

In this chapter, we introduce Aspect Based Sentiment Analysis (ABSA), discuss research motivation, and define four subtasks in ABSA.

1.1 Background

With the boom of Web 2.0 and e-commerce, the way of buying and selling has been greatly changed, especially in retail markets. Knowing what others like and dislike is important for consumers' decision making. In the past, apart from passively watching advertisements, people have to ask friends or turn to authorities in magazines to decide what products are worth buying. The advance of the Web technologies has brought the proliferation of user-generated reviews of products or services on the web. It has become a common practice for people to read the opinions and experiences from the enormous number of people, who are neither their friends nor professional analysts. More and more people are making comments and shopping online, which greatly impacts on the business of both online and off-line business. According to several reports and surveys of more than 5,000 online shoppers [Ste16, Hol16],

- 51% of the purchases of the surveyed shoppers are made on the web;
- 66% of shoppers research online;
- 4.2-4.5 star ratings are the most influential.

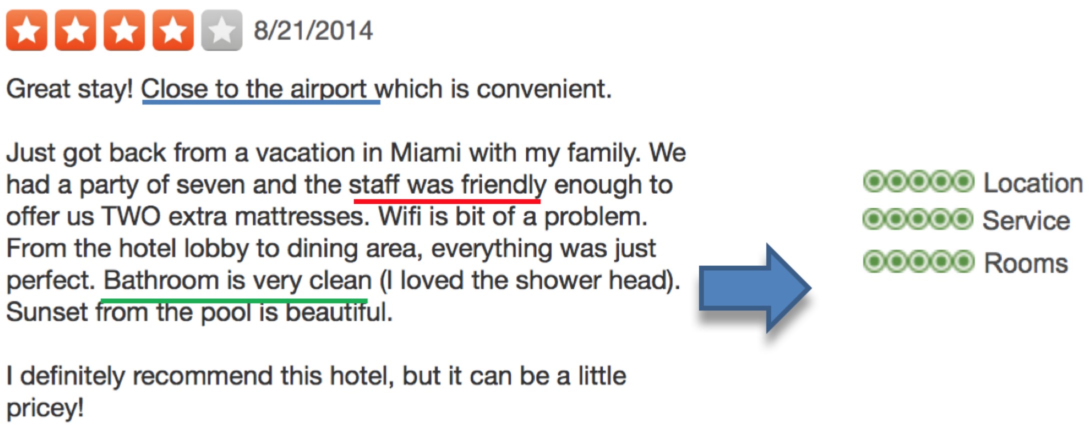
On Yelp and Amazon, two biggest review web hosts, people can further rate comments by the button of "helpful". The most helpful reviews are often ranked higher than others in review sections.

The retailers care more than ever about the reviews left by online customers, not only because the potential influence that opinions can wield in shaping others' purchase intentions [dLFL16], but also because the online comments contribute a valuable resource of feedback for them to further improve their services or products. Monitoring and tracking reputations requires new technologies for content analysis [O'C10]. Therefore, apart from individuals who often browse various reviews before purchasing online or even off-line, vendors and sellers are also eager to see a system that can automatically analyzing consumer sentiment in reviews, rather than clicking and reading pages by pages.

1.2 Aspect Based Sentiment Analysis

Sentiment analysis [PL08, LZ12] is a comprehensive research topic about people's opinion, appraisals, attitudes, and emotions toward entities. Since early 2000, sentiment analysis has been a research focus of many spotlights: natural language processing, web data mining, social media research, consumer research, or even stock market analysis. In the early stage, detecting overall sentiment at document level or at sentence level is the focus of sentiment analysis, rather than at aspect level [Tur02, PLV02, DC04, DLP03, LLS03, MYTF02, NY03, PL08]. Each document can be classified as either positive or negative, in which the whole document is considered as the atomic input.

However, classifying documents by the overall sentiment polarity conceals details [LZ12] that may be critical for recommendation engines, question answering systems, and other applications. Customers often have different preferences to different features of products or services. Those who only skim the average ratings without actual reading paragraphs would omit important information or even be misled. In fact, the overall sentiment polarity can be decomposed into fine-grained sentiment polarities at aspect levels. A positive review towards an entity does not necessarily mean that the author gives positive ratings on all



8/21/2014

Great stay! Close to the airport which is convenient.

Just got back from a vacation in Miami with my family. We had a party of seven and the staff was friendly enough to offer us TWO extra mattresses. Wifi is bit of a problem. From the hotel lobby to dining area, everything was just perfect. Bathroom is very clean (I loved the shower head). Sunset from the pool is beautiful.

I definitely recommend this hotel, but it can be a little pricey!

Figure 1.1: A hotel review example. The original review gave the hotel 4-star general rating. ABSA can provides a deep insight into the 4 stars: critical expressions, aspects, and aspect sentiments.

aspects of the entity. More often than not, positive and negative opinions are mixed at aspect level, sometimes even at sentence level.

Instead of modeling opining mining as a single classification problem, aspect based sentiment analysis (ABSA) actually consists of a series of tasks which can systematically capture and summarize reviews into a fine-grained representative format of the entity. A typical review from TripAdvisor is as follows. TripAdvisor is the biggest travel and restaurant website hosting hotel and restaurant reviews.

In Figure 1.1, the overall rating (4 stars) indicates the reviewer's general impression towards the target hotel; however, without reading sentence by sentence, it is impossible to understand the reasons why the reviewer is satisfied with the hotel, on what aspects, and to what extent. It is entirely possible that the reviewer holds different attitudes towards different aspects of the hotel. People who come across this review have to literally read the review to answer questions about fine-grained sentiments. It is usually inefficient and boring. Imagine that most reviewers simply give superficial comments in just two or three sentences. Similar problems challenge data analysts when they confront restaurant review data. Although a few websites provide options to let reviewers summarize their attitudes in terms of numerical ratings at aspect level, but most of review data needs further process to infer detailed sentiment polarities.

In summary, without fine-grained analysis, we cannot uncover detailed opinions – the specific sentiment polarity, emotion or attitude towards various attributes of an entity – which are often required by recommendation systems, search engines, and potential customers. To this end, applying ABSA fulfills the urgent demand of opinion mining. It can quickly understand the rapidly growing reviews by extracting fine-grained insights such as expressions, aspects, and sentiment polarities [LZ12, PGP$^+$14, PGP$^+$15, PGP$^+$16].

1.3 Problem Definitions

In this section, we first define a few concepts, and describe four tasks in aspect based sentiment analysis, based on SemEval 2016 Task 5 [PGP$^+$16]. It is necessary to clarify the concepts involved in this dissertation, because different names are used interchangeably among many references and data resources, which actually refer to same terminologies.

Definition 1.3.1 (Entity) *An entity is the target that a review comment about. It can be a place, a service, or a product. In this work, we assume that each review in the study of sentiment analysis must have one target.*

Definition 1.3.2 (Aspect) *An aspect is a specific attribute or property of an entity that reviewers may mention. For example, in hotel reviews, an aspect could be "location", "service", and so on. It is a high-level concept and is defined separately from review data. Usually, the vocabulary is rather small.*

Definition 1.3.3 (Aspect Terms) *A sequence of word tokens is a linguistic expression in the review about some features of the entity. It must appear in the given sentence, and usually have a large vocabulary. It is also called target expression.*

The sashimi is always fresh and the rolls are innovative and delicious.
The restaurant was expensive, but the menu was great.

Table 1.1: A restaurant review

An aspect can be either predefined or extracted from the actual data. It is a general facet of the target entity. Specific aspects can be explicitly mentioned in the review, or can be implicitly implied in a intricate way. While, a target expression is a short word phrase that must be explicitly mentioned in reviews. For example, in a restaurant review, the entity of the review is the restaurant. The aspects could include *price*, *food* and so on. The target expressions could be *sushi rolls* or *salad with a delicious dressing*. The two definitions can be represented by same words, but they are defined at different concept levels.

The mission of ABSA is to identify the aspects of entities and the sentiment on each aspect. To fulfill that, we need to decompose ABSA into several subtasks, which can extract aspects, aspect terms, and sentiment polarities [PGP$^+$14, PGP$^+$15, PGP$^+$16].

- **Aspect Extraction (AE).** This task is to learn aspects in a collection of review data, which is formulated as an unsupervised learning task. For example, in Table 1.1, the aspect categories mentioned in the review are “*food*” and “*restaurant*”.
- **Aspect Category Classification (ACC).** This task aims to classify the given text according to a predefined set of aspect categories. It is a special case of text classification.
- **Aspect Term Extraction (ATE).** This task is also referred to target expression detection [PGP$^+$14]. It is dedicated to find out interesting text in a given sentence, which could be a single word or a phrase. For example, in Table 1.1, the aspect terms include “*sashimi*”, “*rolls*”, “*restaurant*”, and “*menu*”.
- **Aspect-level Sentiment Polarity Classification.** Sentiment polarity classification can be performed at many levels: document level, sentence level and aspect level. If

> sentiment polarities are available at aspect level, readers would not have to get bogged in too much detail, but could get more sentiment information than just looking at overall polarity scores. Therefore, the aspect-level analysis is worth of investigation on review data. The given text could be classified as positive, negative, or neutral; or they could be labeled in terms of numerical ratings.

A common strategy for these subtasks is to build a rich set of features and use off-shelf models. However, ABSA is not necessarily a sequential project in which each subtask should be performed one after another. Each task can be done individually, or can be combined with others. In this dissertation, we explore multi-task approaches for solving these subtasks, because each subtask in ABSA is closely correlated. The features learned by one task could be used to guide the other learning tasks. For example, extracted aspect terms in ATE task could be good features for the aspect category classification in AE task. The common feature representation shared by ACC task and ATE task can reduce noise from each task, which brings improvement on robustness.

1.4 Summary and Roadmap

In this dissertation, we focus on developing effective machine learning models based on topic models and neural networks to solve the aforementioned ABSA tasks on review data.

The rest of the dissertation is organized as follows: Chapter 2 proposes several probabilistic graphical models for aspect extraction and sentiment polarity classification at aspect level. Chapter 3 summarizes the recent advance in opinion mining using neural networks. Chapter 4 describes a multi-task learning method for aspect category classification and aspect term extraction. Chapter 5 introduces gated convolutional neural networks for aspect based sentiment polarity classification. Chapter 6 concludes the dissertation with future work.

CHAPTER 2

Topic Modeling: Aspect Extraction and Rating Inference for Hotel Reviews

2.1 Introduction

The trend that people browse hotel reviews on websites before booking encourages researchers to analyze this valuable social media data, i.e., reviews. In a typical scenario, users write down their own opinions and rate hotels with numerical scores. Sometimes, the scores include several aspect ratings predefined by websites such as `room`, `service`, and `location`. The overall rating score expresses a general impression of the reviewer. Although people can understand how the reviewer think about the hotel at first glance, but the overall score hide a lot of details. For example, given a review with 3 stars, it is likely that the reviewer holds different attitude towards different aspects. Without fine-grained analysis, we cannot tell whether the user express negative or positive on what aspects, because the detailed sentiments are mixed into the general overall scores. On the other hand, users usually do not have the patience to read through the review text. To this end, the aspect-specific sentiment analysis provides a good solution. There is a lot of reviews missing aspect ratings. Identifying aspect and learning more informative aspect ratings is an attractive topic in opinion mining. It helps users gain more details of each aspect easily.

Many approaches have been proposed towards aspect-based opinion mining. A comprehensive survey [ME11, ME12] indicates that when using opinion phases, topic model based methods outperform other bag-of-words based models. In Interdependent LDA (ILDA) [ME11], the vocabulary of a collection of reviews is decomposed into two sets: the head terms and the modifier terms with POS Tagging processing. Each review is assumed to be made of several pairs of heads and modifiers. For example, the phrase “nice service” is parsed into a pair of the head term “service” and the modifier term “nice”. The modifier term is used to infer the sentiment polarity, while the associated head terms are the features for aspect identification. The head terms do not have sentiment polarity. Both of the head terms and the modifier terms are modeled as observed variables and conditioned on the latent variables, i.e., rating variables and topic variables. In addition, it is straightforward to consider the dependencies between the rating variables generating the modifier terms and the topic variables producing the head terms, because reviews usually have different preferences across different aspects.

However, the topic models [ME11, ME12, WLZ10] cannot gain any benefit from the available aspect ratings associated with reviews. Aspect ratings are now very easy to be obtained from websites like TripAdvisor and Orbitz. TripAdvisor has the largest volume of reviews: 570 million reviews by 2017. Most reviews are associated with aspect ratings. The problem of traditional topic models is that they do not explicitly model the observed aspect ratings from data. Motivated by this observation, we propose two new topic models which can simultaneously learn aspects and their ratings by utilizing the numerical aspect ratings. Our model can be applied to any review data set without aspect ratings. The aspect ratings are only needed for training. Specifically, our models are based on opinion phrases which are pairs of head and modifier terms. The dependencies between aspects and their ratings are captured by their latent variables. We use Gibbs sampling to estimate the parameters of the models on the training data set and use maximizing a posteriori (MAP) method to predict aspect ratings on unrated reviews.

A preliminary version of the work has been published in [XLR15]. In this journal submission, in addition to revising and elaborating the original paper, we propose new topic model ARIH (Aspect and Rating Inference using Hotel specific aspect rating priors), which extends the prior models and achieves better experiment performance. The rest of paper is organized as follows. Section 2.3 formulates the problem and notation we use. Section 2.4 proposes our model and describes the inference methods. Section 2.4.8 shows the data, the experiments and discuss experiment results. Finally we draw the conclusion and provide future research tasks in Section 2.5.

2.2 Related Work

The problem of review sentiment mining has been an attractive research topic in recent years. There are several lines of research. The early work focuses on the overall polarity detection, i.e., detecting whether a document expresses positive or negative. The author of [PLV02] found that the standard machine learning techniques outperform human on the sentiment detection. Later, the problem of determining the reviewers sentiment with respect to a multi-point scale (ratings) is proposed in [PL05]. The problem was transformed into a multi-class text classification problem. Hidden Markov Model (HMM) is specially adapted to identify aspects and their polarity in Topic Sentiment Mixture model (TSM) [MLW^{+}07]. Ranking methods are also used to produce numerical aspect scores [SB07].

In the literature, Latent Dirichlet Allocation (LDA) [BNJ03] based methods play a major role because the ability of topic detection of LDA is very suitable for multi-facet sentiment analysis on reviews. MG-LDA [TM08b, TM08a] (Multi-Grain Latent Dirichlet Allocation) considers a review as a mixture of global topics and local topics. The global topics capture the properties of reviewed entities, while the local topics vary across documents to capture ratable aspects. Each word is generated from one of these topics. In their later work, the authors modeled the aspect rating as the outputs of linear regressions, and combine them into the model in the corresponding aspect. Joint sentiment/topic model (JST) [LHLN15, LH09] focuses on aspect identification and ratings prediction without any rating information available. In JST, the words of reviews are determined by the latent variables of topic and sentiment. Aspect and Sentiment Unification model (ASUM) [JO11] further assumes all the words in one sentence are sampled from one topic and one sentiment. CFACTS model [LSM14] combines HMM with LDA to capture the syntactic dependencies between opinion words on the sentence level. Given overall ratings, Latent Aspect Rating Analysis (LARA) [WLZ10] uses a probabilistic latent regression approach to model the relationships between latent aspect ratings and overall ratings. On the other hand, the POS-Tagging technique is frequently used in the detection of aspect and sentiment. The authors of [LZS09] categorized the words in reviews into head the terms and the modifier terms with simple POS-Tagging methods. They proposed a PLSI based model to discover aspects and predict their ratings. Interdependent LDA model (ILDA) [ME11] captures the bi-direction influence between latent aspects and ratings based on the preprocessing of head terms and modifier terms. Senti-Topic model with Decomposed Prior (STDP) [CLSZ13] learns different distributions for topic words and sentiment words with the help of basic POS-Tagging. Similar ideas are applied to separate aspects, sentiments, and background words from the text [ZJYL10].

Our models are based on opinion phrases [LZS09], but overcome the drawback of previous models that cannot take advantage of the available aspect ratings. We consider the relationships between several factors, such as overall ratings, aspect ratings, head terms, and modifier terms.

2.3 Problem Formulation

In this section, we introduce the aspect-based opinion task and list notations we use in our models. Formally, we define a data corpus of N review documents, denoted by $\mathcal{D} = \{x_1, x_2, \ldots, x_D\}$. Each review document x_d in the corpus is made of a sequence of tokens. Each review x_d is associated with an overall rating r_d, which takes an integer value from 1 to $S (S = 5)$. An aspect is a predefined property of a hotel, such as `value`, `room`, `location`, and `service`. A text review expresses the reviewer's opinions on several aspects. For example, the occurrence of the word `price` indicates the review comments on aspect `value`. Each review is associated with several integer scores called ratings $\{l_1, l_2, \ldots, l_K\}$, where K is the number of aspects.

Phrase: We assume each review is a set of opinion phrases f which are pairs of head and modifier terms, i.e., $f = \langle h,\ m \rangle$. In most cases, the head term h describes an aspect and the modifier term m expresses the polarity of the phrase. The basic NLP techniques like POS-Tagging are used to extract phrases from raw text for each review.

Aspect: An aspect is a predefined attribute that the reviewers may comment on. It also corresponds a probabilistic word distribution over the vocabulary in the topic models, which can be learned from data.

Rating: Each review contains an overall rating and several aspect ratings. The rating of each review is an integer from 1 to 5. We assume that the overall ratings are available for each review, but the aspect ratings are available only in the reviews used for training.

Review: A review is represented as a bag of phrases, i.e., $x_d = \{f_1, f_2, \ldots, f_M\}$.

Problem Definition: Given a collection of reviews, the main problem is to 1) identify aspects of reviews, and 2) infer the unknown aspect ratings on the unrated reviews.

2.4 Proposed Models

In this section, we propose two generative models to solve the aspect-based opinion mining task by incorporating observed aspect ratings. We list the notations of the models in Table 2.1. We assume reviews are already decomposed into head terms and modifier terms using NLP techniques [ME11].

D	the number of reviews
K	the number of aspects
M	the number of opinion phrases
S	the number of distinct integers of ratings
U	the number of head terms
V	the number of modifier terms
z	the aspect / topic switcher
l	the aspect rating
h	the head term
m	the modifier term
r	the overall rating
θ	the topic distribution in a review
π	the aspect rating distribution for each topic
α	the parameter of the Dirichlet distribution for θ
β	the global aspect sentiment distribution
λ	the parameter of the Dirichlet distribution for β
δ	the parameter of the Dirichlet distribution for ϕ and ψ
ϕ	the head term distribution for each topic
ψ	the modifier term distribution for each sentiment

Table 2.1: The table of notations

2.4.1 Assumptions

We discuss some assumptions in modeling review text. First, our models presume a flow of generating ratings and text. The reviewer gives an overall rating based on his experience, then rates the hotel on some aspects and writes down review text. In the model of bag-of-phrases, the reviewer chooses a head term for an aspect on which he would like to comment, then he picks a modifier term to express his opinion. This generation process is captured by our models.

Second, the aspect ratings depend on the overall ratings. For example, when a user gives a 5-star overall rating, it is unlikely that the user gives low ratings on any of the aspects. An average overall score indicates the reviewer is disappointed on some aspects, but not all of them. It is possible that the reviewer holds positive feedbacks on other aspects. Inspired by this observation, we model the aspect ratings π with multinomial distributions $P(\pi|r)$ conditioned on the overall rating r.

Third, the aspect ratings imply another relationship with modifier terms of opinion phrases [ME12]. Because, for different aspects, people use different words to express different attitude. For example, it does not make any sense to use the word “patient” to comment on the aspect “room”. We explicitly introduce random variables for modifier terms which are conditioned on aspect variables, so that meaningful aspects and sentiments can be learned from the head and the modifier terms respectively.

2.4.2 Motivation

Existing topic models do not require aspect ratings of reviews during model training and consider it as an advantage. It may be true in the past, since there are not many reviews containing aspect ratings. Nowadays, more review hosts, such as TripAdvisor and Orbitz, allow reviewers to rate hotels on predefined aspects. The volume of such extended reviews is growing rapidly. It is reasonable to leverage the valuable information to build more precise and accurate models. To the best of our knowledge, this study is the first work to utilize the aspect ratings.

Our topic models assume aspect ratings as probabilistic variables. The aspect ratings π are scores in reviews on K aspects. They are available in the training data and hence treating them as switchers is quite straightforward. An interesting observation is the distinction between the aspect rating and the phrase sentiment. They are both sentiment switchers and are conditioned on the overall rating variable r. One is for the aspect, the other is for the phrase. If we assume that both of them are generated from the prior aspect sentiment distribution β and the overall rating r, we have ARID model (Aspect and Rating Inference with the Discrimination of aspect sentiment and phrase sentiment). The interaction between π and r is through the global β and the overall rating r. It saves the direct dependency between them. If we assume in given the aspect k, the reviewer holds the same sentiment for all the modifier terms, the discrimination between aspect sentiment and phrase sentiment becomes redundant and can be removed. The model ARIH model (Aspect and Rating Inference using Hotel specific aspect rating priors) extends the prior model (ARIM) in our work [XLR15]. We consider the prior probabilistic distribution β of aspect ratings for each hotel. It allows the aspect ratings of reviews to be sensitive to the hotel own characteristics.

2.4.3 ARID Model

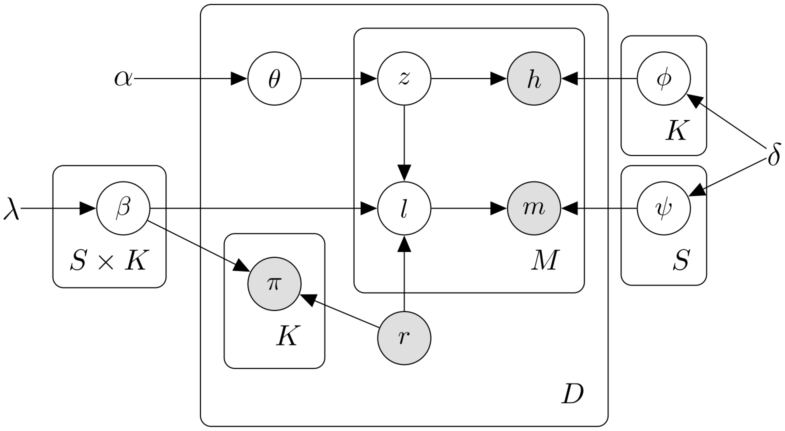


Figure 2.1: Graphical Representation of ARID model. The outer box represents D reviews, while the inner box contains M phrases

ARID model, shown in Fig. 2.1, captures the review generation process and the two dependencies described above. Following the conventional topic models for review analysis, we use random variables z and l to simulate the generating process of the head and the modifier terms respectively. The topic selection variable z is governed by a multinomial topic distribution θ. The sentiment variable l for each opinion phrase is determined by the aspect sentiment variables β, the overall rating r and the aspect switcher z.

Specifically, in ARID model, the variables π representing aspect ratings are shaded in the graphical representation since they are observed in the training dataset. They become latent variables for prediction on unrated reviews. The latent sentiment variable l is sampled from β_k where k is determined by the value of z. The overall rating variable r serves as a prior variable for both the aspect rating π and the phrase sentiment l.

The formal generative process of our model is as follows, where Dir denotes Dirichlet distribution and Mult denotes Multinomial distribution.

- For each aspect k and each overall rating value of r
  - Sample the aspect sentiment distribution $\beta_{r,k} \sim \texttt{Dir}(\lambda)$
- For each review x_d,
  - Sample latent topic distribution variable $\theta_d \sim \texttt{Dir}(\alpha)$
  - For each aspect k from 1 to K in the review,
    * Sample aspect rating $\pi_{d,k} \sim \texttt{Mult}(\beta_{r_d,k})$
  - For each phase i from 1 to M in the review,
    * Sample aspect indicator $z_i \sim \texttt{Mult}(\theta_d)$
    * Sample sentiment indicator $l_i \sim \texttt{Mult}(\beta_{r_d,z_i})$
    * Sample head term $h_i \sim \texttt{Mult}(z_i, \phi)$
    * Sample modifier term $m_i \sim \texttt{Mult}(l_i, \psi)$

2.4.4 ARIH Model

In this section, we improve a previous model. The new model ARIH (Aspect and Rating Inference using Hotel specific aspect rating priors) is shown in Figure 2.2. Like the previous model ARIM (Aspect and Rating Inference Merging aspect sentiments and phrase sentiments) [XLR15], we assume the aspect sentiment is equivalent to the phrase sentiment. In other words, the modifier terms that belong to one aspect share the same sentiment, i.e., the aspect sentiment. Therefore, we can use only one polarity indicator for both the aspect and the phrase. In particular, the aspect ratings π are modeled as in ARID, but π also indicates the phrase sentiment. Since the aspect ratings are available in the training data,

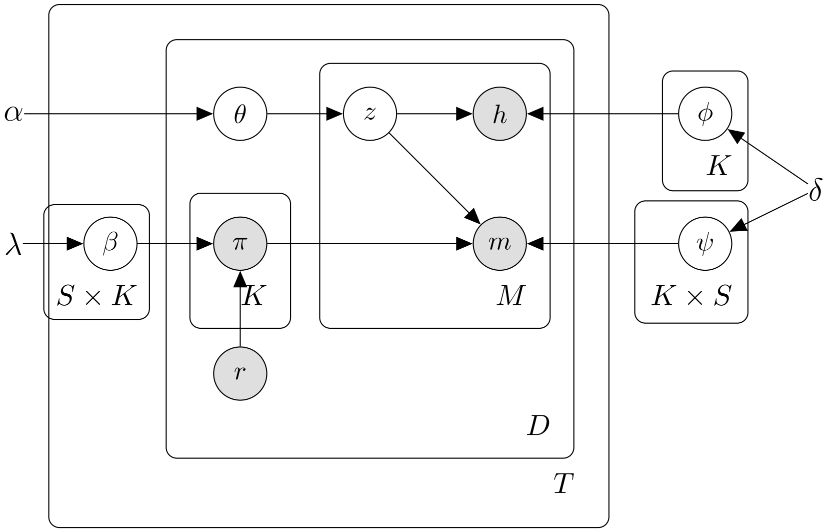


Figure 2.2: Graphical Representation of ARIH model

the information from β to m is blocked by π according to the d-separation theory of graphical models [Bis07]. Therefore, the modifier term is determined by the aspect ratings π instead of β, and the aspect variable z. In the generative procedure of ARIH, the modifier term m_i is sampled from $\psi_{z_i,\pi_{z_i}}$. π follows a multinomial distribution with parameter β.

In ARID and ARIM, we assume the aspect rating variables π is conditioned on the global aspect sentiment distributions β, which has the size of $K \times S$. For each aspect k and each global rating s, we have a Dirichlet distribution over ratings β. However, making the aspect rating conditioned on the global aspect sentiment distributions ignores the aspect rating biases of different hotels. Each hotel has its own pros and cons. For example, despite the hotels may have the same global rating, the one located near the airport would receive higher ratings on `location`, while those having good service may be rated higher on `service`. If both of them get 4-star ratings, the aspect ratings π have the same global prior distribution $\beta_{k,s=4}$.

To verify our assumption, we use Principle Component Analysis (PCA) to investigate the distribution of β. In particular, for each hotel, we compute the average ratings on each aspect, which give the same overall rating. It generates a matrix P, where $P_{i,j}$ is the average

overall rating	1	2	3	4	5
largest variance ratio	0.704	0.753	0.831	0.871	0.965

Table 2.2: The largest variance ratio of principle component analysis

rating of the jth aspect of the ith hotel. We have five P matrices for each possible overall rating ranges from 1 to 5. We use PCA to reduce the dimension of P and compute the largest variance ratio in Table 2.2. The variances of aspect ratings are quite large, especially for 5-star overall rating. If the variance is 0.871, for example, the ratings on some aspect can be 1-star difference for difference hotels. The analysis shows that the same overall ratings often imply different weights on aspects, which depend on the type of hotels.

Different from ARIM, ARIH associates each hotel with its own aspect rating priors $\beta_{t,r,k}$. Here, $\beta_{t,r,k}$ represents the aspect rating distribution on aspect k when the overall rating is r for hotel t. ARIH extends ARIM by using hotel-specific beta, therefore ARIM can be considered as a special case of ARIH. If we apply ARIH on the collection of reviews of one hotel. ARIH is reduced to ARIM. In Figure 2.2, the graphical model ARIH has one more layer than ARIM. The variables for each review in ARIH model have one more subscript to indicate which hotel the review comes from.

2.4.5 Estimation

ARID Model

There are two methods widely used for parameter estimation, i.e., Gibbs sampling [GS04] and variational inference [BNJ03]. Since updating equations using Gibbs sampling is relatively easy to derive and implement, we adopt collapsed Gibbs sampling (CGS) which integrates out the intermediate random variables θ, ϕ, β, and ψ. For prediction, we learn the distributions ϕ, ψ of the head and the modifier terms as well as the global aspect sentiment distribution β from z and l. The Gibbs sampling repeatedly samples latent variables $z_{a,b}$ and $l_{a,b}$ conditioned on all other latent z and l in document a for phrase b.

In ARID model, the joint probability is

$$\begin{aligned} p(z, l, h, m|\alpha, \lambda, \delta, \pi, r) = \int & p(\theta|\alpha)p(z|\theta)\times \\ & p(h|z, \phi)p(\phi|\delta)\times \\ & p(\pi|\beta, r)p(l|\beta, r, z)p(\beta|\lambda)\times \\ & p(m|l, \psi)p(\psi|\delta)\; d\theta\, d\beta\, d\phi\, d\psi \;, \end{aligned} \tag{2.1}$$

where we integrate out θ, ψ, β and ϕ.

We define two counters $N_{d,r,k,s,u,v}$ and $C_{d,r,k,s}$ to count the numbers of the occurrences of opinion phrases $f_{d,i} = \langle h_{d,i} = u,\; m_{d,i} = v\rangle$ and the aspect rating $\pi_{d,k}$. Specifically, $f_{d,i} = \langle h_{d,i} = u,\; m_{d,i} = v\rangle$ is the phrase i of document d which has the head term u and the modifier term v. $N_{d,r,k,s,u,v}$ is the number of times that the pair of head term u and modifier term v is assigned to aspect k and sentiment s in document d, whose overall rating is r. $C_{d,r,k,s}$ is the indicator of the document d that gives aspect rating s on aspect k when the overall rating of the document is r. Although given document d, its overall rating r_d is determined, we use the overall rating as a subscript for convenience.

$$N_{d,r,k,s,u,v} = \sum_{i=1}^{M} \mathbf{I}[r_d = r,\; z_{d,i} = k,\; l_{d,i} = s,\; h_{d,i} = u,\; m_{d,i} = v] \;, \tag{2.2}$$

$$C_{d,r,k,s} = \mathbf{I}[r_d = r,\; \pi_{d,k} = s] \,, \tag{2.3}$$

where the function $\mathbf{I}$ is the identify function. We replace the subscript N by $*$ when summing out the counter along the subscript indices. For example,

$$N_{d,r,*,s,u,v} = \sum_{k=1}^{K} N_{d,r,k,s,u,v} \;. \tag{2.4}$$

Gibbs sampling samples $z_{a,b}$ and $l_{a,b}$ simultaneously

$$\begin{aligned}p(z_{a,b}, l_{a,b}|z_{-(a,b)}, l_{-(a,b)}, \alpha, \delta, \lambda, h, m, r, \pi) \propto\ & (N^{-(a,b)}_{a,r_a,z_{a,b},*,*,*} + \alpha)\times \\ & \frac{N^{-(a,b)}_{*,*,z_{a,b},*,h_{a,b},*} + \delta}{N^{-(a,b)}_{*,*,z_{a,b},*,*,*} + U\delta}\times \\ & \frac{N^{-(a,b)}_{*,r_a,z_{a,b},l_{a,b},*,*} + C_{*,r_a,z_{a,b},l_{a,b}} + \lambda}{N^{-(a,b)}_{*,r_a,z_{a,b},*,*,*} + C_{*,r_a,z_{a,b},*} + S\lambda}\times \\ & \frac{N^{-(a,b)}_{*,*,*,l_{a,b},*,m_{a,b}} + \delta}{N^{-(a,b)}_{*,*,*,l_{a,b},*,*} + V\delta}\ .\end{aligned} \tag{2.5}$$

It turns out that the aspect ratings π can be considered as pre-observed phrase sentiment counts for the global aspect sentiment distribution β. Therefore, the prior parameter λ can be dropped. We estimate the aspect sentiment distribution β with the aspect ratings π and the overall ratings r of the training data before Gibbs sampling with Equation (2.6).

$$\beta_{r,k,s} = \frac{C_{*,r,k,s}}{C_{*,r,k,*}}\ . \tag{2.6}$$

The third term of the right hand of Equation (2.5) is replaced by

$$\frac{N^{-(a,b)}_{*,r_d,z_{a,b},l_{a,b},*,*} + \tilde{\lambda}\beta_{r_d,z_{a,b},l_{a,b}}}{N^{-(a,b)}_{*,r_d,z_{a,b},*,*,*} + \tilde{\lambda}}\ , \tag{2.7}$$

where $\tilde{\lambda}$ is the scaling factor for β. The parameters of ARID ψ, ϕ, θ are estimated by

$$\phi_{k,u} = \frac{N_{*,*,k,*,u,*} + \delta}{N_{*,*,k,*,*,*} + U\delta},\ \psi_{s,v} = \frac{N_{*,*,*,s,*,v} + \delta}{N_{*,*,*,s,*,*} + V\delta},\ \theta_{d,k} = \frac{N_{d,r_d,k,*,*,*} + \alpha}{N_{d,r_d,*,*,*,*} + K\alpha}\ . \tag{2.8}$$

ARIH

The iterative updating function of Gibbs sampling for ARIH has little difference from that for ARID.

$$\begin{aligned}p(z_{a,b}|z_{-(a,b)}, \alpha, \delta, \lambda, h, m, r, \pi) \propto\ & (N^{-(a,b)}_{a,r_a,z_{a,b},*,*,*} + \alpha)\times \\ & \frac{N^{-(a,b)}_{*,*,z_{a,b},*,h_{a,b},*} + \delta}{N^{-(a,b)}_{*,*,z_{a,b},*,*,*} + U\delta}\times \quad . \\ & \frac{N^{-(a,b)}_{*,*,z_{a,b},\pi_{a,z_{a,b}},*,m_{a,b}} + \delta}{N^{-(a,b)}_{*,*,z_{a,b},\pi_{a,z_{a,b}},*,*} + V\delta}\end{aligned} \tag{2.9}$$

The parameters of ARIH model ϕ, θ is estimated by Equation (2.8), but the number of ψ is $K \times S$, which is estimated by

$$\psi_{k,s,v} = \frac{N_{*,*,k,s,*,v} + \delta}{N_{*,*,k,s,*,*} + V\delta} \ . \tag{2.10}$$

We estimate β by

$$\beta_{t,r,k,s} = \frac{C_{t,r,k,s}}{C_{t,r,k,*}} \ , \tag{2.11}$$

where $C_{t,r,k,s}$ counts the number of the reviews which assign the overall rating r and the aspect rating s on the aspect k for hotel t.

2.4.6 Incorporating Prior Knowledge

We use a small set of seed words to initialize the aspect term distribution ϕ [WLZ10]. Without any prior knowledge, we have to set the number of topics and align the generated aspects with predefined aspects. It is neither necessary nor easy for analyzing hotel reviews, because we are interested in only a few widely-used aspects. We consider the seed words as the pseudo-counts, i.e., the amount of δ words are added to $\phi_{k,u}$ before Gibbs sampling.

2.4.7 Prediction

The goal of our models is to predict aspects and ratings on the unrated reviews. Given an opinion phrase $f_{d,i} = \langle h_{d,i},\ m_{d,i}\rangle$ and the overall rating r_d in a new document d, we identify the aspect on which the phrase $\hat{z}_{d,i}$ comments and predict the aspect rating $\hat{l}_{d,i}$.

We use CGS to sample z and l together from $p(z, l|h, m, r, \alpha, \beta, \phi, \psi)$, where θ is integrated out. Here, two subscripts d and i are dropped for simplicity. After enough sampling iterations, we first estimate the predicted aspect $\hat{z}$ by the most frequent z among the pairs of $\langle z,\ l\rangle$. It is equivalent to use MAP (Maximum a posteriori estimation) by integrating out l. Then given the predict $\hat{z}$, we predict $\hat{l}$ with $\mathbb{E}[p(l|\hat{z}, h, m, r, \beta, \phi, \psi, \alpha)]$.

The reason why we consider the expectation of l is that the aspect ratings are numerical, rather than independent discrete category labels. The probability of each possible value l are kind of importance. The aspect mixture weight θ for a new document can be learned by Gibbs sampling as well, but we simply assume θ is a uniform distribution, because a review on hotel should comment on all the concerned aspects.

When ARIH is applied on the reviews without aspect ratings, we integrate out the latent aspect rating variable π and θ, then sample z from $p(z|h, m, r, \alpha, \beta, \phi, \psi)$ to compute MAP $\hat{z}$, like ARID model. For each opinion phrase $\langle h, m \rangle$ whose $\hat{z} = k$, we assign the most probable sentiment score $\hat{s} = \arg\max_s \psi_{\hat{z},s,m}$ to the modifier term m. Then, the estimated aspect rating $\mathbb{E}[p(\pi_k|\hat{z}, h, m, r, \beta, \phi, \psi, \alpha)]$ is computed by averaging the scores $\hat{s}$ of all the opinion phrase whose $\hat{z} = k$.

2.4.8 Experiments

In this section, we describe the review data we use and evaluate the performance of our models.

Data and settings

The data set we use is crawled from TripAdvisor [WLZ10]. Each review in the data set is associated with an overall rating and 7 aspect ratings, which are within the range from 1 to 5. However some aspects such as `Cleanliness`, `Check in / front desk` are rarely rated. To better train and evaluate models, we use only four mostly commented aspects, `Value`, `Room`, `Location` and `Service`. We keep reviews with all four aspect ratings to evaluate the models. We use NLTK [1] to tokenize the review text, remove stop words, remove infrequent words, apply POS-Tagging technique [ME11] to extract opinion phrases, and filter out short reviews which contains less than 10 phrases. The final data set contains 1,814 hotels and

[1] http://www.nltk.org

aspect	seed words
Value	value, fee, price, rating
Room	windows, room, bed, bath
Location	transportation, walk, traffic, shop
Service	waiter, breakfast, staff, reservation

Table 2.3: Seed words

aspect	head terms	modifier terms
Value	deal, price, charge	good, great, reasonable
Room	house, mattress, view	comfortable, clean, nice
Location	parking, street, bus	great, good, short
Service	manager, check-in, frontdesk	friendly, good, great

Table 2.4: Frequentest head terms and modifier terms

31,013 reviews. We randomly take 80% of data as the training data set, the rest of them as the test data set. 10-fold cross validation is used to tune the hyper-parameters α and β on the training data set. The seed words used to initialize the head term distribution ϕ is in Table 2.3, which is a small set of words.

Aspect Identification

In this section, we demonstrate that the ability of identifying meaningful aspects. Since the head terms found by the two models are not so different from each other, we present top 3 frequentest head terms for each aspect in Table 2.4. The listed head terms are the most frequent words, which have highest values in ϕ_k. We also list top 3 frequentest modifier terms for each aspect. The models can successfully extract ratable aspects from reviews and learn aspect-specific sentiment words as well. For example, "comfortable" is frequently used to describe aspect "Room", but not for other aspects. We also observe that people also like to use vague sentiment words for all aspects, such as "good", "great".

Metric

We use RMSE(Root-mean-square error) [2] to measure the performance of predicting aspect ratings for each hotel in the test set. Assuming the predicted aspect rating for hotel d on aspect k be $\hat{\pi}_{d,k}$ and ground-truth $\pi_{d,k}$, RMSE is represented as Equation (2.12).

$$\text{RMSE}(\hat{\pi}_{d,k}, \pi_{d,k}) = \sqrt{\frac{1}{DK}\sum_{d=1}^{D}\sum_{k=1}^{K}(\hat{\pi}_{d,k} - \pi_{d,k})^2} \tag{2.12}$$

RMSE measures the accuracy of the prediction on aspect ratings. We also use Pearson correlation in Equation (2.13) to describe the linear relationship between the predicted and the ground-truth aspect ratings. Here, π_d is the vector of the aspect ratings of document d.

$$\rho_{\text{aspect}} = \frac{1}{D}\sum_{d=1}^{D}\rho(\pi_d, \hat{\pi}_d) \tag{2.13}$$

Since the rating score is an ordinal variable, we adopt Pearson linear correlation ρ_{aspect} on the aspect ratings within each review to evaluate how a model keeps the aspect order in terms of their scores. For each aspect, we also compute the linear correlation across all hotels ρ_{hotel} as in Equation (2.14). The measure is used to test whether the model can predict the order of hotels in teams of an aspect rating. π_k consists of all the aspect ratings of all the hotels on the aspect k.

$$\rho_{\text{hotel}} = \frac{1}{K}\sum_{k=1}^{K}\rho(\pi_k, \hat{\pi}_k) \tag{2.14}$$

Aspect Rating Prediction

We present the experiment results on the reviews without any aspect rating in Table 2.5. We compare the results between our models and one baseline. The baseline predicts all the aspect ratings of each review with the given overall ratings. The baseline predicts the aspect ratings of a review with a constant value, so $\rho_{\text{aspect}} = 0$. The results indicate that ARID and ARIH(ARIM) outperform the baseline and LARAM [WLZ10]. The main reason is that our

[2] http://en.wikipedia.org/wiki/RMSE

measure	baseline	LARAM	ARID	ARIM	ARIH
RMSE	0.702	0.632	0.573	0.505	0.481
ρ_{aspect}	0.0	0.217	0.185	0.259	0.328
ρ_{hotel}	0.755	0.755	0.737	0.764	0.781

Table 2.5: Performance of Aspect Inference

models can capture the dependency between the aspects, the aspect ratings and the modifier terms, by taking into the account the aspect ratings in the training data set. In terms of ρ_{hotel}, all the approaches have similar scores. On the hotel level, the aspect ratings are averaged across all reviews, while the goals of these four methods are predicting the ratings of each individual review. The difference between each method on predicted aspect ratings for each review is small. Therefore, there is no much difference on the measure ρ_{hotel}.

Moreover, ARIH (ARIM) is better than ARID, which confirms our observation. The sentiment of aspects and modifiers is not much different from each other. Reviewers express the same polarity with different modifier terms, when commenting on one aspect. Therefore, merging aspect sentiment with modifier sentiment does not decrease the capability of the models. ARID model has K kinds of modifier term distributions ψ, while ARIH has $K \times S$, since the modifier term m in ARIH is dependent on the aspect switcher z and the aspect sentiment π. ARID estimates a global sentiment distribution across all aspects, while ARIH can learn aspect-specific sentiment distribution by modeling aspect-dependent sentiment. In the inference, the aspect on which the opinion phrases comment is determined by its head term h. ARID infers the polarity for each modifier term from a coarse sentiment distribution, while ARIH can obtain more find-grained sentiment using its $K \times S$ modifier term distributions. The parameter ψ in ARIH fine-tunes the predicting results based on β and ϕ. Therefore, in terms of Pearson correlation metric, ARIH has better performance than ARID.

ARIH model has more aspect rating distributions β than ARIM. It gives better accuracy on predicting the polarity of the modifier terms. The difference between the aspect ratings

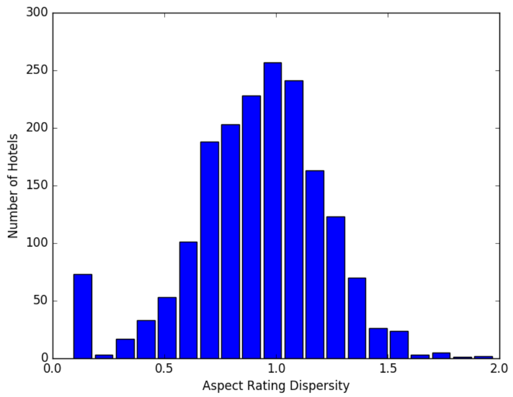


Figure 2.3: Aspect Rating Dispersity of Hotels

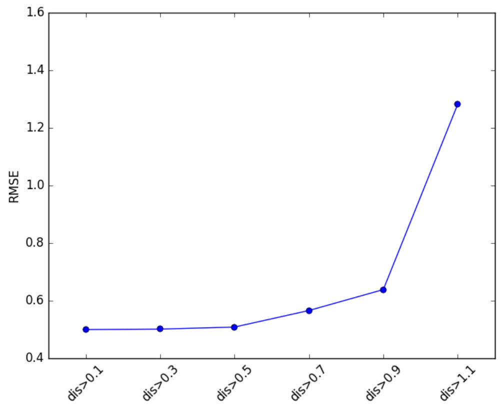


Figure 2.4: Aspect Rating Prediction on Reviews of Different Dispersities

in β and those in reviews may influence the performance. Following the experiments in [LZC$^+$14], we investigate the relationships between the dispersity and RMSE of ARIH. The dispersity is given by Equation (2.15), where the we take the mean value of β and compare it with the aspect ratings of reviews for each hotel. As displayed in Figure 2.3, for most hotels, the aspect ratings dispersity are around 1.0 and have a clear Gaussian distribution. Moreover, there are some hotels having 0 dispersity, since the highest rating score is 5. There is very little gap between the aspect ratings and the averaged ones for highest-rated hotels.

$$\text{dis} = \sqrt{\frac{\sum_{i=1}^{K}(\mathbb{E}[\beta_{t,k}] - \pi_{t,k})^2}{K}} \tag{2.15}$$

As Figure 2.4, we randomly prepare data set with different dispersity and report RMSE of ARIH on them. “dis >0.1” means that we use the reviews which has dispersity larger than 0.1. ARIH performs well on reviews with dispersity lower than 1.3. Due to the small training data and the high variance of aspect ratings on reviews with large dispersity, the performance of ARIH decreases.

2.5 Conclusion

In this chapter, we propose two models for aspect identification and sentiment inference, ARID and ARIH. They utilize the overall ratings and the aspect ratings in reviews to identify the aspects and uncover the corresponding hidden aspect ratings. The models are based on topic models, but explicitly consider the dependency between the aspect ratings, the aspect terms, and sentiment terms. The opinion phrases which consist of head terms and modifier terms are extracted by simple POS-Tagging techniques. The most important contribution is that the models incorporate the aspect ratings as observed variables into the models and significantly improve the prediction performance of aspect ratings. The difference between them is that ARIH merges the sentiment variables of the modifier terms with those of the aspects. ARIH further considers the hotel-specific aspect rating priors β. Gibbs sampling and MAP is used for estimation and inference respectively. The experiments on large hotel review data set show that the models have better performance in terms of RMSE and Pearson correlation. In the future, we would investigate the methods that can automatically generate ratable aspects from the text, not from the predefined seed words. Another interesting research topic is to explore the relation between different aspects [GX13], because the different aspects in one review may share the similar sentiments.

CHAPTER 3

Deep Learning Models for Aspect Based Sentiment Analysis

3.1 Introduction

Sentiment analysis is an interdisciplinary research field involving natural language processing, web data mining, social media research, and consumer research. Text sentiment analysis requires techniques in natural language processing, such as named entity recognition to recognize target items of interest, polarity analysis to determine whether a review expresses positive attitude, and syntax analysis on words and sentences. It also needs information retrieval techniques about data storage and search for opinions, images, and videos that users contribute to websites. The impressing needs of personalized recommendation systems brings up other challenges, like accurately identifying the patterns of user behaviors in online shopping, and recommending goods and services by tracking user preferences.

It is important to point out that sentiment analysis is a part of various applications and systems. A typical example is review-oriented website, in which sentiment analysis is essential to the website. For example, Yelp hosts more than 142 million reviews, which provide most of the content. The main function is to collect and distribute user generated reviews and ratings. People can access a vast pool of quantitative and qualitative feedbacks provided by other users on hotels, restaurants, or other facilitates. When uploading reviews, users are asked to rate the target place on a five point scale. Reviewers have an option to add photos to support their reviews. Visitors who read reviews can help the website to rank existing reviews by clicking buttons of “useful”, “funny”, and “cool”. Using techniques of sentiment analysis, the website can fulfill users’ demands for fast digesting and analyzing large-scale collections of comments, such as summarizing a lot of reviews by aggregating and highlighting well-received items in the majority of reviews, or providing personalized rankings of restaurants according to the past reviews of the user. In recommendation systems, the personalized recommendation results can be enhanced by considering sentiment polarities of reviews at fine-grained level [PL08].

In this chapter, we focus on text sentiment analysis using deep learning techniques. According to the taxonomy of sentiment analysis [YSZ17a], it consists of two subcategories: opinion mining and emotion mining. Opinion mining studies the attitude towards some entities implied in text expression; while emotion mining is about the feeling of authors. We only cover recent research effort with deep learning concentrates on opinion mining. In 2008, Pang et al.[PL08] gave a comprehensive survey on opinion mining, which covers various tasks such as feature extraction, entity recognition and document summarization, but the methods in that survey heavily rely on feature engineering. In 2012, another survey by Liu et al. [LZ12] studied topic of sentiment and subjectivity classification, aspect based sentiment analysis, and opinion spam detection. In 2017, Yadollahi et al. [YSZ17a] focused on the text sentiment analysis and summarized recent work about opinion mining and emotion mining. However, neither of them cast light on the hot trend of deep learning methods. In 2018, Zhang et al. [ZWL18] also summarized deep learning models on sentiment analysis, but in this chapter we cover most recent research work and focuses on review data as well as various tasks.

Sentiment analysis [PL08, LZ12] is a comprehensive research topic about people's opinion, appraisals, attitudes, and emotions toward entities. We use the review from SemEval 2016 Task 4 [PGP$^+$16] and walk through an example to introduce a big picture. SemEval workshop defines several subtasks at aspect level, and the released datasets are popular benchmarks for aspect based sentiment analysis.

Let's look at a concrete review example.

1. *Judging from previous posts this used to be a good place, but not any longer.*
2. *We, there were four of us, arrived at noon - the place was empty - and the staff acted like we were imposing on them and they were very rude.*

3. *They never brought us complimentary noodles, ignored repeated requests for sugar, and threw our dishes on the table.*
4. *The food was lousy - too sweet or too salty and the portions tiny.*
5. *After all that, they complained to me about the small tip.*

When reading these reviews, we notice that they express a general negative attitude to the restaurant. Sentences (1) gives a negative opinion on general impression. In sentence (2) and (3), the reviewer is disappointed by the service. In sentence (4), the quality of food further makes the review feel that he should not be there. Again, sentence (5) states a discomfort experience about service again. Certainly, the overall sentiment polarity is negative. We also can identify different targets in different sentences. Sentence (2), (3) and (5) are about service, while the aspect of sentence (4) is food. Besides analyzing the overall sentiment, aspect based sentiment analysis introduces several subtasks at fine-grained level, which would be discussed on Section 3.

The surge of interest in sentiment analysis on review drives researchers to develop new technologies based on machine learning and data mining. Researchers develop automatic tools to distill opinions, detect sentiment polarities, and summarize user-generated text on the same topic. Conventional machine learning methods heavily depend on the representation of data to discover the mapping from features to targets. Feature engineering is as critical as modeling. Handcrafted features usually require a large amount of domain expertise and human labor. If one trains a classic supervised learning method such as SVM or Bayesian classification to predict the sentiment polarity of a movie review, he needs to engineer a set of discriminative features. The traditional methods heavily relied on following techniques [PLV02]:

Frequency Features These features are commonly used in information retrieval and opinion mining. A vocabulary of terms is built in the first place, which is used to index the terms that appear in the dataset. For each document, a high-dimension vector encodes the presence/frequency of all terms according to the dictionary. The Tf-idf weighting is a default option for many tasks.

N-gram Features A n-gram is a sequence of adjacent words in a document. Recent study [WM12] shows that bi-gram features gives consistent performance gains on sentiment classification, when Naive Bayes or SVM classifier is used.

Part-of-Speech Features A part-of-speech feature of a word indicates useful grammatical properties. For instance, in sentiment polarity classification, adjectives are good indicators of polarity in most cases. While nouns are most likely to be aspect terms. The rich linguistic features are also widely used in named entity recognition and extraction of aspect terms [RR09].

Lexicons The words in lexicons express either positive or negative sentiments in most scenarios. They are collected manually or by unsupervised learning [Tur02]. The semantic orientation of such words are estimated by computing pointwise mutual information.

Negations Definitely, negation words such as "not" may reverse the polarity of a sentence. It is a common practice to attach "NOT" to the adjective near negation words to create a new word token. However, some exceptions like "no wonder" needs careful treatment.

Syntactic Dependency The tree of syntactic structure of a sentence could be used to derive seeds and rules for aspect term extraction [QLBC11].

Recently, deep learning has enabled the representation learning within end-to-end deep neural networks. The main idea of deep learning was re-branded with many names [GBC16], until recent revival in 2006 [HOT06]. The idea of deep learning went through several ups and downs since 1940s [GBC16]. The key concepts, such as distributed representation [HS], back-propagation [RHW], long short-term memory (LSTM) [HS97] and convolutional neural network (CNN) [LBBH98], still remain the essential ingredients of today's deep learning models. However, they did not receive much attention as they do today, until Geoffrey Hinton introduced deep belief network in 2006, a generative graphical model which could be efficiently trained by layer-wise pre-training [HOT06]. Since then, deep learning has been reshaping the landscape of many related research domains, such as computer vision, natural language processing, speech recognition and reinforcement learning [LHB15].

The availability of large-scale labeled datasets and advances of computational power contribute to the revival of deep learning thoughts developed decades ago. The dataset size grows exponentially, from MNIST [LBBH98] which consists of 70 thousand scanned hand-written digits, to ImageNet [DDS+09] having 1.2 million images of 1000 categories, and to the most recent Youtube-8M [AEHKL+16] composed of 8 million annotated videos (total 500k hours). Deep learning develops in the era of Big Data. It is still an open question that how to leverage large amount of data, especially, most of which are unlabeled data from industry with unsupervised learning.

Large models are possible to be trained with large datasets, but they cannot live without massive computational power. Fast graphical processing units (GPUs) has been introduced [RMN09] to machine learning, which brings incredible speedup to model training. Large clusters and dedicated servers like Nvidia DGX-1 have been built for trainig larger and larger deep learning models. Tensor processing units (TPUs), which have even up to 70x faster than GPUs and more power efficiency, have been used in Google datacenter and Google Cloud [JYP+17].

Apart from the advent of faster hardware, the growth of deep learning is also driven by better software infrastructure, including CUDA [NBGS08], Theano [BBB10], Caffe [JSD+14], Torch [CK11], Tensorflow [AAB+16], Pytorch (http://pytorch.org), Caffe2 (http://caffe2.ai), and many others. Thanks to the persistent contributing of open source communities, it becomes easy to develop deep learning models on advanced computing hardware without dealing with low level programming.

The capacity of the neural models is increasing dramatically in terms of the number of layers or the number of neurons. For example, in computer vision, the number of layers of modern neural networks starts from 8 layers (AlexNet [KSH12]), 16 layers (VGGNet [SZ15]), 22 layers (GoogleLeNet [SLJ$^+$15]) to astonishingly 152 layers (ResNet [HZRS16a]); while the error rates on ILSVRC image classification task dropped drastically.

In conclusion, the renaissance of deep learning can be contributed to large-scale data, powerful models, and increased computational resources. Deep learning methodology has been sweeping over many research areas. Driven by this evolution, this survey provides an overview of existing methods proposed for sentiment analysis and opinion mining using deep learning. To make this survey self-contained, a short introduction to neural networks and essential modules is provided first. Afterward, we present the architectures of neural networks employed for the important tasks in sentiment analysis and opinion mining.

3.2 Basic Modules in Deep Learning

3.2.1 Word Embeddings

For many deep learning applications in natural language processing, word embeddings are cornerstones for deep neural networks. However, the concept of word embedding is not new. The idea could date back to 1950s, "*You shall know a word by the company it keeps*" proposed by Firth [Fir57]. It is also called distributional semantic model (DSM) or distributed representation [Hin84], because it is assumed that the meaning of a word can be learned by its distribution in text. Today, the word embeddings are represented in terms of high-dimension vectors which can capture semantic and syntactic information from text.

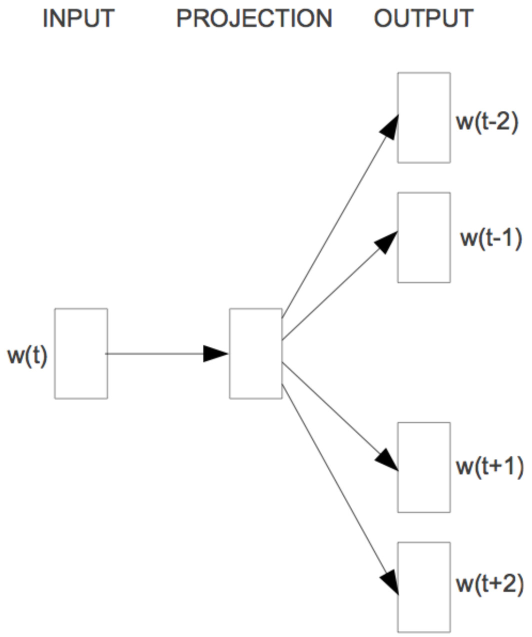


Figure 3.1: The diagram of skip-gram [MCCD13]

Early DSMs, such as probabilistic Latent Semantic Analysis [Hof99] and Latent Dirchellet Analysis [BNJ03], have been developed in 1990s, followed by neural network based embedding models such as distributed representation learning for words [BDVJ03, HSMN12] and multi-tasking embedding learning model SENNA [CW08]. The prevalence of word embedding begins CBOW [MCCD13, MSC$^+$13] and GloVe [PSM14] in 2013 and 2014.

Word2vec is arguably the most popular word embedding model in recent years. It includes two architectures CBOW and skip-gram. Take skip-gram for an example, as shown in Figure 3.1. The objective function of skip-gram sums the log likelihood of the surrounding words given the target word in a slide window:

$$\frac{1}{N}\sum_{i}\sum_{j\in n(i)}\log p(w_j|w_i)$$

Skip-gram does not have hidden layers as early neural network based models [BDVJ03]. Therefore,

$$p(w_j|w_i) = \frac{\exp(v'^{\top}_{w_j} v_{w_i})}{\sum_k \exp(v'^{\top}_{w_k} v_{w_i})} \quad ,$$

where w_i and w_j are the word embedding vectors that would be trained by gradient descent. An efficient training involves dozens of approximation and training techniques [MSC+13].

In many NLP applications, most neural networks have embedding layers, in which the words are embedded to a lower-dimension space and can be directly trained during back-propagation training. However, the embedding vectors are by-products and are limited to the availability of data in the application domain. Methods such as word2vec explicitly aim to generate word embeddings and can explore massive unlabeled data. Therefore, in most cases, the parameters of the embedding layers are initialized with pre-trained word embeddings, so that the down-stream task can leverage the semantic relationships from a large data corpus. For example, one version of GloVe pre-trained word vectors is trained on a data set of 840 billion tokens.

Although both word2vec and GloVe claim superior performance over traditional DSMs in many tasks, a recent study [LG14] unveiled that word2vec implicitly factorizes a word-context matrix of Pointwise Mutual Information (PMI). It means all of these models share similar nature: matrix decomposition. More detailed analysis [LGD15] isolates various factors that account for the success of neural networks based embedding models.

3.2.2 CNN

CNN has been an essential architecture for deep learning researchers. It has a long history, which dates back to the study of cells in animal visual cortex in 1959 [HW59]. In 1990s, LeCun et al. invented and improved first practical CNN, LeNet-5 [LBD+89, LBBH98], for hand-written digit recognition. It did not gain much attention compared with support vector machine [CV95] and probabilistic graphical models [KF09] at that time, until the tremendous success of AlexNet [KSH12] in ILSVRC 2012. Large training data and computational resource to CNN is what aviation gasoline to jet engines. The represen-

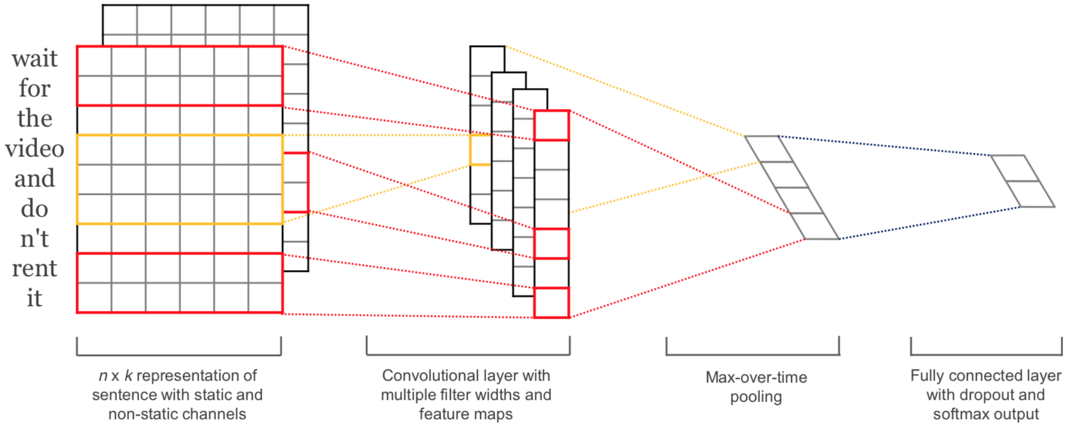

wait
for
the
video
and
do
n't
rent
it
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 3.2: The diagram of CNN for text classification [Kim14]

tative variations inspired by AlexNet include VGGNet [SZ15], GoogleNet [SLJ$^+$15] and ResNet [HZRS16a].

A convolutional layer contains multiple kernels (or filters) to compute feature maps. Specifically, unlike fully-connected layers where each neuron is connected all neurons of previous layers, a convolutional neuron is only connected to a few neurons in a small region called receptive field (local connections). Afterwards, an element-wise activation function is applied. Suppose the input feature is a matrix $\mathbf{X}$, a 2D convolutional neuron computes a new feature at location (i, j) as

$$z_{i,j} = f\left(\sum_{m=-k/2}^{k/2} \sum_{n=-k/2}^{k/2} \mathbf{X}_{i+m,j+n} \mathbf{W}_{m+k/2+1,n+k/2+1} + b \right) \quad .$$

It is often written as $\mathbf{Z} = \mathbf{X} \odot \mathbf{W} + b$, where $\odot$ denotes the convolutional operation. b is a scalar bias and f is a non-linear activation function, such as ReLU [NH10] or tanh function. A max-pooling layer takes the maximum of values in a region.

$$y_{i,j} = \max\{z_{m,n}\}, \quad (m, n) \in \mathcal{R}_{i,j} \quad ,$$

where $\mathcal{R}$ is a local region around location (i, j), which is usually a square region. Multiple convolutional layers and pooling layers can be stacked together in a special way. Finally, there are several fully-connected layers on top of them.

The key ideas of CNN are local connections, shared weights, pooling and the use of many layers [LHB15]. All the neurons computing one kind of feature map share only one set of convolution weights. The weight sharing reduces the complexity of the model as well as the number of parameters. It corresponds with an intuition that a good enough detector should be applied widely across a large area.

There are a number of options for activation functions. In the early days, people just use sigmoid and tanh functions. The gradient vanishing problem of tanh function is mitigated by LeCun's tanh function [LBOM12]. Later, in AlexNet [KSH12], Rectified Linear Units (ReLUs) are proposed to speed up calculation and become prevalent in deep learning area. Leaky ReLU [MHN13] allows for a small, non-zero gradient when the unit is saturated at negative inputs. ReLUs would be dead and unable to update their weights when they have negative valued inputs. In Parametric ReLU [HZRS15], the slope of Leaky ReLU can be learned. Maxout [GWFM$^+$13] has very strong fitting ability but would double the number of parameters. Exponential Linear Unit (ELU) [CUH16] combines a linear term on positive inputs and an exponentially decayed term on negative inputs, which prevents the dead neuron problem of ReLU. Scaled Exponential Linear Units (SELU) [KUMH17] is based on ELU but with two fixed scalar parameters, which forms the self-normalizing neural network.

The architecture of CNN also varies. VGGNet [SZ15] and Inception [SVI$^+$16] module use several very small 3x3 convolution filters instead of large ones, which shows a significant improvement with fewer parameters. GoogleNet [SLJ$^+$15] adopts a deep and wide structure which can capture features at multiple granularities. Bottleneck [SVI$^+$16] which consists 1x1 convolution layers before and after 3x3 convolutions further reduces the number of parameters. ResNet [HZRS16b, HZRS16a] has an extremely deep architecture with compelling accuracy and nice convergence behaviors. Xception [Cho16] uses a spatial convolution performed independently over each channel of an input, followed by a pointwise convolution. A more comprehensive survey about CNN could be found in the survey of Gu et al. [GWK+15].

3.2.3 LSTM

Recurrent neural networks (RNNs) [RHW86, Elm90, Gra12] enable sequences learning, which have a memory to summarize previous inputs to influence the network output. Specifically, at every time stamp t, the hidden state $\mathbf{h}_t$ of a RNN is produced by a composition function using the previous hidden state $\mathbf{h}_{t-1}$ and the current input $\mathbf{x}_t$. The non-linear gates determine what to store and what to forget in the hidden state. During the linear composition process, the weight matrices $\mathbf{W}$ are shared across each time stamp. For example, Elman-type recurrent network [Elm90] is defined as

$$\begin{aligned} \mathbf{h}_t &= \sigma(\mathbf{W}^{(x)}\mathbf{x}_t + \mathbf{W}^{(h)}\mathbf{h}_{t-1} + \mathbf{b}^{(h)}) \\ \mathbf{o}_t &= \sigma(\mathbf{W}^{(o)}\mathbf{h}_t + \mathbf{b}^{(o)}) \quad . \end{aligned}$$

Vanilla RNN suffers from the vanishing and exploding gradient when training with back-propagation [BSF94, HS97, PMB13]. Long-short term memory (LSTM) [HS97, Gra12] introduces memory cells and four nonlinear gates which control the information follow through memory cells. It can store and access information over long periods of time,

[0] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

therefore mitigates the vanishing gradient problem.

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}^{(i)}\mathbf{x}_t + \mathbf{U}^{(i)}\mathbf{h}_{t-1} + \mathbf{b}^{(i)}) \\
\mathbf{f}_t &= \sigma(\mathbf{W}^{(f)}\mathbf{x}_t + \mathbf{U}^{(f)}\mathbf{h}_{t-1} + \mathbf{b}^{(f)}) \\
\mathbf{o}_t &= \sigma(\mathbf{W}^{(o)}\mathbf{x}_t + \mathbf{U}^{(o)}\mathbf{h}_{t-1} + \mathbf{b}^{(o)}) \\
\mathbf{u}_t &= \tanh(\mathbf{W}^{(u)}\mathbf{x}_t + \mathbf{U}^{(u)}\mathbf{h}_{t-1} + \mathbf{b}^{(u)}) \\
\mathbf{c}_t &= \mathbf{i}_t \odot \mathbf{u}_t + \mathbf{f}_t \odot \mathbf{c}_{t-1} \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
\end{aligned}$$

Gated recurrent unit (GRU) [CvMG+14] simplifies the updating rules of LSTM and achieves competitive performance. Bi-direction versions of RNN and LSTM [SP97, GS05] are simple but effective extensions which allow models to access information from the past and the future at any time stamp. Multiplicative RNN [SMH11] and LSTM [KLMR16] make the parameters controlling the transition from hidden vectors to hidden vectors to depend on the inputs $\mathbf{x}$, which brings performance improvement in natural language modeling.

3.2.4 Memory Networks and Attention Mechanism

Neural turing machine [GWD14] and memory networks [WCB14] introduce to recurrent neural networks a large external memory, on which a model can read or write. To make the model trainable, it is needed to make such neural networks differentiable with respect to the location in the memory that it reads from or writes to. Neural turing machine uses attention mechanism for addressing, which not specifies a single location but assigns different weights everywhere according the similarities between a query vector and all memories. Then, the output of reading is a weighted sum of all memories; the result of writing is a convex combination of the old memory content and the new value. Memory networks [WCB14, SSWF15] is designed for natural language processing tasks, such as question and answering. It has four operations: I, G, O, and R. I converts the text to an internal feature representation. G updates memories given the new input. O computes output features given the new input and the memory. R decodes the output feature to given the final response.

Attention mechanism has gained popularity in many research areas, such as image captioning [XBK$^+$15], question answering [WCB14], and neural machine translation [BCB14]. In recent study on neural machine translation [BCB14], the sentence in source domain language represented by a sequence of word embeddings is encoded into a fixed-length vector by a LSTM encoder; then it is used to generate a sentence in target language by a LSTM decoder. However, both the encoder and the decoder have to deal with long-range dependency only through a single vector. Instead of solely depending on the single sentence embedding from the encoder, the LSTM decoder equipped with attention mechanism is able to "attend" to necessary parts of inner hidden states of the encoder according to its current generation state and the source sentence embedding. The basic formulation is as follows. Suppose we have a query vector $\mathbf{x}$ and a matrix of feature vectors $\mathbf{H}$, we would like to extract a vector which represents the information of $\mathbf{H}$ which are most related to $\mathbf{x}$. First, an attention mechanism may compute a similarity score between $\mathbf{x}$ and each vector in $\mathbf{H}$ by one of following functions [LPM15]:

$$s(\mathbf{x}, \mathbf{h}_i) = \begin{cases} \mathbf{x}^\top \mathbf{h}_i \\ \mathbf{x}^\top \mathbf{W} \mathbf{h}_i \\ \mathbf{v}^\top \text{tanh}(\mathbf{W}[\mathbf{x}; \mathbf{h}_i]) \end{cases},$$

where $\mathbf{W}$ and $\mathbf{v}$ are parameters for the score functions. Then a softmax function normalize the scores over all the memories:

$$\mathbf{a} = \frac{\exp(s(\mathbf{x}, \mathbf{h}_i))}{\sum_j \exp(s(\mathbf{x}, \mathbf{h}_j))}$$

Finally, we get the representation vector by element-wise production between $\mathbf{a}$ and each $\mathbf{h}_i$:

$$\mathbf{u} = \sum_i a_i \cdot \mathbf{h}_i$$

3.3 Training Strategy

Backpropagation is a critical algorithm used for training deep learning models. Usually, a deep learning model has a loss function $\mathcal{L}$ depending on the outputs of the model with parameter w. We need the partial derivatives $\partial\mathcal{L}/\partial w$ with regard to any parameter w in order to optimize them with gradient descent method. Backpropagation speeds up this process. It makes computing the partial derivatives of the parameters million times faster. Basically, it describes how the error signals from the loss function change and propagate to each parameter, and how to compute the partial derivatives for the current parameters. For a detailed introduction, one could refer to neural network books [GBC16]. Once we get the partial derivatives of parameters, we can update parameters by stochastic gradient descent (SGD). Most deep learning libraries provided automatic differentiation.

At the early stage of deep learning, due to the inadequacy of labeled data and computing resource, neural networks performed poorly that were trained with back-propagation and random initialization. Hinton et al. [HOT06, BLPL06] developed an effective training strategy which consisted of a pre-training stage with unsupervised data and a global fine-tuning stage for training restricted Boltzmann machine. SGD and other variances are commonly used for training most neural networks now. The vanilla SGD performs a parameter update for each training instance x and y.

$$w_{t+1} = w_t - \alpha \nabla f(w; x_t, y_t) \quad , \tag{3.1}$$

where w_t is the t-th time value of parameter w, f is the function about network parameters, α is step size or learning rate. SGD has a high variance because each update is determined by a single data instance. Therefore, mini-batch gradient descent is more frequently used in practice. Mini-batch gradient descent compute the derivatives according to a small batch of data examples, whose size usually ranges from 16 to 256. To dampen the oscillations of updates on non-smooth loss functions, Momentum SGD [Qia99] updates with history gradients, and Nesterov accelerated gradient [Nes83] uses an approximation of future gradient. Learning rate is the most critical hyper-parameter when training deep learning models. An increasingly decayed learning rate is necessary to avoid over-shooting problem near a local minima. Moreover, a model of high depth has gradient vanishing problem result in different learning rates in different layers. Adagrad [DHS11], Adadelta [Zei12], Adam [KB15] and RMSprop are proposed for adaptively adjusting learning rates for each parameter. For a more comprehensive and intuitive visualization of various optimization strategies, readers could refer to the survey [Rud16].

When the objective function is approaching local minimas, the smaller learning rate leads to a slow convergence. Another training strategy is to add normalization layers so that a larger learning rate is affordable. They also alleviate the gradient vanishing problem due to the saturation of some activation functions and the gradient explosion problem because of the structure of recurrent neural networks. The main idea of batch normalization [IS15] is to normalize the inputs at each layer for each mini-batch using the mean and the variance of the inputs within a batch. Layer normalization [BKH16] for recurrent neural networks directly computes the normalization statistics from the summed inputs to the neurons within a hidden layer, so that the normalization does not introduce any new dependencies between training cases within a batch. Dropout and variational dropout is another regularization technique. Dropout [SHK$^+$14] randomly drops neurons from the neural network during training, which significantly reduces overfitting problem. Variational dropout [GG16] is introduced to recurrent neural network based on Bayesian interpretations of dropout.

3.4 Advances in Deep Learning Models on Reviews

Since the tremendous success in computer vision and natural language processing, deep learning models also gradually reshape the landscape of research area on review data. According to a recent survey of sentiment analysis [YSZ17a], sentiment analysis and opinion mining could be categorized into several parts: subjectivity detection, opinion polarity classification, opinion spam detection, opinion summarization, and emotion mining. Review is a special type of social media, by which users express their attitude towards products, service, or places. Therefore, people are more interested in a subset of the subtasks of opinion mining. We classify the existing approaches related to reviews into five main categories in terms of problems they try to solve: (I) polarity classification, (II) aspect term extraction, (III) aspect term polarity classification, (IV) aspect category polarity classification, (V) review summarization and generation.

3.5 Polarity Classification

Classifying the polarity of a given text is a basic but important task in sentiment analysis: whether the expressed attitude in a document, a sentence, an entity, or an aspect is positive, negative or neutral. A large body of work related to reviews falls into this category. After all, the most interesting factor in reviews is how products or services are received by customers. In this section, we discuss opinion polarity classification at different levels. In most existing work, "sentiment analysis" or "sentiment classification" is used as synonyms of opinion polarity classification, although sentiment analysis actually includes many other subtasks. In this chapter, we use the name "polarity classification". In terms of granularity, it can be categorized at document, sentence, entity, or aspect level [YSZ17a].

At document level, the target of polarity classification is the whole review document. The review could be treated as a atomic unit of input or a unit bearing hierarchical structures that consists of paragraphs, sentences, and words. People usually hold different attitudes in different sentences and paragraphs. It may be beneficial to summarize the sentiment polarities from bottom to top according to the internal structure of the given document.

At sentence level, the goal of this task is to determine the polarity of the given sentence. Small structure such as clauses, and the syntax dependencies can be exploited. One of challenges is that user-generated reviews often have grammar mistakes, misspelling errors, or emoticons.

At aspect level, there are two different tasks. One is to predict the sentiment polarity with regard to the specific aspect of a product or service, which belongs to a set of predefined classes. The words of aspect categories may or may not appear in the text. The other is to identify the polarity concerning the interesting entities which appear in text. Most aspect entities are phrases and the size of the vocabulary of aspect phrases could be more than a thousand. For instance, in the sentence "*Average to good Thai food, but terrible delivery.*", the reviewer expresses positive towards the entity `Thai food`, but negative toward the aspect `SERVICE`, which actually does not show in the given text.

Opinion polarity classification is usually defined as a supervised learning problem. One critical step is how to represent text data. For general-purpose machine learning models such as support vector machine and naive Bayes [PLV02], people have to manually construct useful features to encode text into a sparse high-dimension vector. However, neural network based learning models are often based on distributed representation of words, each of which is embedded into a low dimension space. In the next following sections, we classify these neural networks into two categories: neural networks for general text classification and neural networks for sentiment polarity classification on reviews.

3.5.1 Neural Networks for Text Classification

In fact, opinion polarity classification is a special task of text classification, in which the target label is the sentiment polarity of given text. Therefore, neural networks in this category could be applied to other similar problem, such as sarcasm detection, subjectivity classification and emotion classification. We summarize the relevant work in the follow Table 3.5.1.

Table 3.1: The neural networks for text classification

Name	Datasets	Base Model	Highlights
Collobert et al. [CW08, CWB$^+$11]	RCV1, Reuters	CNN	multi-task learning
Kim [Kim14]	MR, SST, Subj TREC, CR, MPQA	CNN	first CNN on text classification
Johnson and Zhang [JZ15]	IMDB RCV1, Elec	CNN	bag-of-words-like CNN
Ma et al. [MHZX15]	MR, SST, TREC	CNN	dependency modeling
dos Santos et al. [dSG14]	SST, STS	CNN	character-level features
Zhang et al. [ZZL15]	AG, Sogou, DB, Yelp, Amazon	CNN	character-level features
Conneau et al. [CSBL16]	AG, Sogou, DB, Yelp, Amazon	CNN	29 convolutional layers
Johnson and Zhang [JZ17]	AG, Sogou, DB, Yelp, Amazon	CNN	15 layers deep pyramid CNN
Le et al. [LCD17]	AG, Yelp, DBpedia, Yahoo	CNN	shallow-and-wide CNN

Socher et al. [SPH^{+}11]	EP dataset	Recurive NN	recursive autoencoders
Socher et al. [SHMN12]	IMDB, MR, SemEval2010 Task8	Recursive NN	matrix-vector recursive NN
Socher et al. [SPWC13]	SST	Recursive NN	recursive neural tensor network
Tai et al. [TSM15]	SST, SICK	Recursive NN	tree LSTM
Chen et al. [CQZ^{+}15]	SST, TREC	Recursive NN	gated recursive neural network on a full binary tree
Qian et al. [QTH^{+}15]	SST	Recursive NN	tag-specific composition function
Liu et al. [LQH17]	SST, MR, TREC, SUBJ, IE	Recursive NN	dynamic composition using meta learning
Iyyer [IMBGDI15]	MR, IMDB, SST		simple but effective
Tang et al. [TQL15a]	Yelp, IMDB	Recurrent NN	hierarchical neural network at document level
Liu et al. [LQH16]	SST, SUBJ, IMDB	Recurrent NN	multi-task learning, shared memory
Xu et al. [XCQH16]	IMDB, Yelp	Recurrent NN	cache mechanisms
Yang et al. [YYD^{+}16]	Yelp, IMDB, Yahoo, Amazon	Recurrent NN	hierarchical, attention mechanisms
Qian et al. [QHLZ17]	MR, SST	Recurent NN	linguistically regularized LSTM

Zhang et al. [ZLR16]	MR, SST, TREC, SUBJ, IMDB	Hybrid	dependency sensitive, LSTMs + Convolution Layer
Wang et al. [WJL16]	MR, SST	Hybrid	RNN + CNN
Wang et al. [WYLZ16]	SST, CVAT	Hybrid	CNN + LSTM
Hsu et al. [HMJS17]	SST, Yelp	Hybrid	phrase focused CNN
Alghunaim et al. [AMCG15]	SemEval	embedding	three subtasks of aspect based sentiment analysis
Li et al. [LZL$^+$16]	IMDB, RT, MPQA, CR, SUBJ	embedding	neural bag-of-n-grams model
Le and Mikolov [LM14]	SST, IMDB	embedding	distributed representations of sentences and documents
Kiros et al. [KZS$^+$15]	MR, CR, SUBJ, MPQA, TREC	embedding	encoder-decoder framework
Zhang et al. [ZRW16]	SUBJ, SST, TREC, Irony	CNN	multi-group norm constraint CNN
Yu et al. [YWLZ17]	SST	embedding	refine word embeddings for sentiment analysis

Convolutional Neural Networks

Convolutional neural networks are originally invented for computer vision and have been shown to be effective for various NLP problems. Early work using CNN for NLP appeared in [CW08, CWB$^+$11]. The authors defined a unified convolutional neural architecture for multiple tasks in natural language processing that learns features without task-specific feature engineering. The NLP tasks include part-of-speech tagging, chunking, named entity recognition, and semantic role labeling. All of these tasks are integrated into a single general convolutional network, which is jointly trained. The authors demonstrated that joint modeling can improve generalization performance. It is also shown that how the combined tasks learn powerful features in the absence of hand-engineered features.

Kim [Kim14] showed a simple CNN on top of the pre-trained word vectors for sentence-level classification tasks. The architecture consists of: a pre-trained word embedding layer, a convolutional layer with many filters, a max-over-time pooling layer that extracts the maximal values over the whole sentence [CWB$^+$11], and a fully connected softmax layer predicting the probabilities of class labels. The authors experimented a variant with "multi-channel" of word embeddings, in which the word vectors in one channel is kept static and the other is fine-tuned by back-propagation training. Despite little tuning of hyper-parameters, a simple CNN with one layer of convolution performs remarkably well. Meanwhile, it suggests that the pre-trained vectors are "universal" feature extractors that can be used for various tasks.

Dynamic Convolutional Neural Network (DCNN) [KGB14] is another convolutional neural network. The layers in the network interleave one-dimensional convolutional layers and dynamic k-max pooling layers. Dynamic k-max pooling returns the subsequence of k maximum values in the input sequences, instead of the single maximum value in classic max-pooling layer. At lower convolutional layers, multiple one-dimensional filters [Kim14] convolve with input sequences to extract n-gram features at every position. At higher layers, the convolutional filters can learn syntactic relations between words that are far apart in the input sentence.

A comprehensive sensitivity analysis of such one-layer CNN is represented in [ZW15], in which the authors empirically investigated the effectiveness of various architectures of

CNN and hyper-parameters on several benchmarks, The settings include word embedding layer, filter region size, activation function, pooling strategy, and so on.

Johnson and Zhang [JZ15] considered that the word embedding layer is a special case of convolution with region size one. They directly applied convolutional neural network to high-dimensional one-hot vector of word regions without going through word embedding layer. Therefore, the proposed bow-CNN is a simple CNN having bag-of-word conversion in the convolution layer.

Ma et al. [MHZX15] proposed a very simple dependency-based convolutional neural networks (DCNNs), based on Kim's CNN model [Kim14]. In Kim's CNN model, the sequential convolutions process the input words in the order of the input sentence. Ma et al. considered a word and its parent, grand-parent, great-grand-parent, and siblings on the dependency tree. In every position, the input words in a convolutional window are rearranged according to the new order derived from the dependency tree. By this way, DCNN captures long-distance information that are unavailable on classic CNN model.

CNN could also be directly applied on characters instead of words. It has better tolerance of misspelling and grammar mistakes, which are quite often in reviews and tweets. dos Santos et al. [dSG14] proposed a deep convolutional neural network, called Character to Sentence Convolutional Neural Network (CharSCNN), which exploits both word-level embeddings and character-level embeddings in a convolutional neural network for sentiment analysis of short texts. The idea of using convolutional neural networks to extract from character-level features inspired many other works.

In the work of Zhang et al. [ZZL15], a character-level convolutional neural network for text classification was described without word-level embedding. The neural network accepts a sequence of characters encoded by character quantization. The encoded information goes through 6 convolutional layers and 3 fully-connected layers. Dropout layers are inserted between fully connected layers to regularize the model. Data augmentation using English thesaurus is employed to control the generalization as well. It not only shows that the syntactic or semantic structure is not required for text classification, but also shows that deep convNets do not require knowledge of words.

Conneau et al. [CSBL16] believed that hierarchical representations of whole sentence can be obtained with deep architectures, which can lead to better performance. They presented a very deep architecture for text classification which operates directly at character level. Inspired by ResNet [HZRS16b, HZRS16a] and VGGNet [SZ15] in computer vision, the authors used up to 29 convolutional blocks and shortcut connections in their model. Every convolutional block is a sequence of two convolutional layers, each of which is made of a temporal BatchNorm layer [IS15] and an ReLU activation [NH10]. It shows "benefit of depths" for convolutional neural networks in NLP.

Although character-level approaches have strong points in not having to deal with a large vocabulary of words, a later work [JZ17] shows that a knowledge of word leads to a powerful representation. The authors proposed an effective and efficient design of deep word-level CNNs for text classification. The deep pyramid CNN model simply interleaves a convolution block and a downsampling, and the internal data size shrinks in a pyramid shape. Therefore, at higher level, the discovery of long-range dependencies is more efficient.

In computer vision, people have acknowledged the importance of depth in neural networks. However, it is still in debate what level is the best for the model to based on and whether very deep architecture is better for text classification. An extensive experimental study on several text classification and sentiment analysis [LCD17] shows that shallow-and-wide CNN at word level is more effective and that the increasing depth of CNN does not bring significant performance improvement.

Recursive Neural Networks

Recurrent neural networks treat text as a flat sequence of words or characters regardless of hierarchical structure of natural language. However, a significant amount of research work has been done about recursive neural networks, which take such structure information into account.

Recursive autoencoder (RAE) was proposed by Socher et al. [SPH$^+$11]. Autoencoder is a simple three-layer neural network learning the compressed representation of the inputs. RAE recursively uses autoencoders to compose word embedding vectors and the outputs of last composition. RAE does not rely on the predefined tree-like structure over the given sentences to determine the order of composition, but learn it in a greedy manner by minimizing the loss of reconstruction of autoencoders. After this unsupervised learning stage, a semi-supervised setting with an additional cross-entropy error is set to learn the sentiment distribution of the given sentence.

Socher et al. [SHMN12] proposed Matrix-Vector Recursive Neural Network (MVRNN) later, in which each constituent (a word or longer phrase) in a tree is represented by a pair of a vector and a matrix. The vector captures the meaning of that constituent; while the matrix captures how it modifies the meaning of the other word that it combines with. The matrix-vector representations for constituents are computed bottom-up by recursively combining the words according to the syntactic structure of a parse tree. The composition functions are non-linear with respect to the parameters, input word vectors and matrices. By adding on top of each node a softmax classifier, MVRNN can be trained for sentiment polarity classification with back-propagation. The author demonstrates the ability to capture semantic compositionality in a syntactically plausible way, which leads to improved performance on learning sentiment distributions.

Later, Socher et al. [SPWC13] introduced a sentiment detection data set called Stanford Sentiment Treebank (SST), in which syntactically plausible phrases in all sentences are manually labeled. This dataset later becomes a standard benchmark of sentiment polarity classification. A new network called Recursive Neural Tensor Network (RNTN) is presented. Like MVRNN, RNTN computes the vector representation of a phrase through a parse tree and tensor-based functions, and then computes vectors for high nodes in the tree using the same composition function. It captures the compositional effects with high accuracy. Compared with MVRNN, RNTN uses a composition function containing tensor production between parameters and input vectors, which reduces the number of parameters without losing representation ability.

Tree-LSTM [TSM15] is a generalization of LSTM to tree-structured network topologies. Like RNTN and MVRNN, the structure of Tree-LSTM is built according to the results of external syntactic parsers. While the standard LSTM sequentially composes its hidden state from the input at the current time stamp and the hidden state in the previous stamp; Tree-LSTM composes its states from an input vector and the hidden states of many child nodes. Two variations are considered: Dependency Tree-LSTM and Constituency Tree-LSTM. When the word embedding vectors are initialized with GloVe vectors [PSM14], Constituency Tree-LSTM has better accuracy on the Stanford Sentiment Treebank [SPWC13].

In the work of Gated Recurrent Neural Network (GRNN) [CQZ$^+$15], the pre-processing stage to construct the tree structure is not needed. Instead, a full binary tree structure is employed with neural gates to control the composition in recursive structure. Like GRU [CGCB14], GRNN also has two kinds of gates: reset and update gates to control the combinations within the binary tree structure. Although its performance on SST and QC is not as good as the other recursive neural networks which reply on the external syntactic parsers, it may be robust on noisy text data.

Qian et al. [QTH$^+$15] thought that the composition function for the syntactically different phrases should be different. The proposed tag guided RNN (TG-RNN) requires extracted syntactic information and uses the syntactic tag of the parent phrase to control the composition process from the child nodes. Therefore, they way to compose noun phrases is distinct from that to compose adjective phrases. Meanwhile, TG-RNN also learns the embedding vector for each phrase tag, which is concatenated with phrase vector as input to composition function. As a result, the phrases that have informative tags could contribute more to sentiment expression.

Liu et al. [LQH17] also studied the limitation in the richfulness of compositionality in recursive neural network. Different parameters should be used for different kinds of semantic compositional functions. They proposed a dynamic compositional neural networks over tree structure. In this model, a meta network maintains the shared meta-knowledge across different positions of compositions and dynamically generates the context-dependent compositional functions. Therefore, the composition functions of the current inputs vary according to the contexts of the inputs.

Iyyer et al. [IMBGDI15] believed that the learning of recursive neural networks is computational expensive. They presented efficient deep averaging network (DAN), where the unordered compositions are learned on word embeddings without syntactic tree. It obtains near state-of-the-art accuracies on a variety of sentence- and document-level sentiment classification tasks. DAN works in three simple steps: takes the vector average of the embeddings of the input tokens, passes the average vector through one or more feed-forward layers, and performs linear classification on the final layer's output.

LSTM and Memory Networks

On document-level polarity classification, it is worth utilizing the hierarchical structure: words, sentences, and documents. LSTMs can be naturally stacked together at different levels, which are usually coupled with attention mechanisms.

Tang et al. [TQL15a] assumed that the overall sentiment of a document depends not only on the sentiments of individual sentences, but also on the relationships between sentences in the semantic meaning of document. In this hierarchical neural network, the sentence representations from word embedding are built by LSTM or CNN. Then, a gated recurrent neural network is used to adaptively encode semantics of sentences and their complex linguistic relationships in document representations.

Liu et al. [LQH16] extended LSTMs with an external memory maintained by reading and writing operations as in Neural Turing Machine [GWD14]. The external memory can store long term information and knowledge, which could be shared by several related tasks. The fusion gates control the information flow between an external memory and LSTMs, and selectively utilize the shared information. The authors aligned two extended LSTMs for multi-task learning on different data sets. The experimental results show that jointly learning of related tasks improves the performance in terms of sentiment classification accuracy on movie reviews and product reviews.

Xu et al. [XCQH16] also extended LSTMs with a cache mechanism to capture the long-range sentiment information. The internal memory is divided into several groups with different forgetting rates. Different groups capture different-scale dependencies by changing the forgetting rates. This enables the ability of capturing the local and global emotional information, thereby better summarizing the sentiment polarity of the given document.

Hierarchical attention networks [YYD$^+$16] modeled documents with a hierarchical structure which consists of word-level and sentence-level bi-directional GRUs. First, the representations of sentences are built by a GRU layer. Their outputs are then aggregated into a document representation by another GRU layer. Second, different words and sentences in a document are differentially informative and highly context dependent. GRU layers include two levels of attention mechanisms. The probabilities of document classes are computed with a fully-connected layer with aggregated hidden vectors of the sentence-level GRU cells.

Qian et al. [QHLZ17] developed a simple sequence model that fully employs linguistic resources to benefit sentiment classification task. Three types of resources are addressed: sentiment lexicons, negation words, and intensity words. The central idea is to regularize the difference between the predicted sentiment distribution of the current position and that of the previous or next position on a bidirectional LSTM for a sentence-level sentiment classification task.

Hybrid models

Hybrid models combine two or more different types of neural networks together. Bradbury et al. [BMXS16] studied the limitation of RNNs and tried to parallelize RNN. Quasi-recurrent neural (QRNN) model combines convolutional neural network and recurrent neural network. Instead of relying computation results on th previous times-step, QRNN uses parallel convolution operations instead of linear mapping in vanilla RNN. All the gates were no longer dependent on previous states of RNNs. Three types of pooling functions with different configurations of gates are discussed.

Wang et al. [WJL16] proposed a jointed CNN and RNN architecture for sentence-level sentiment classification. It takes local features extracted by CNN as the input to RNN. The max-pooling layer of a vanilla convolutional architecture is replaced with a LSTM layer, whose final hidden state is then used to predict the sentiment of the whole sentence.

Wang et al. [WYLZ16] presented another hybrid model for document-level sentiment analysis in which CNN is employed for sentence modeling, whose outputs are fed to a document-level LSTM to generate a pair of scores for the whole document. The scores are in valence-arousal space, in which the dimension of valence measures the degree of positive and negative sentiment, while the dimension of arousal refers to the degree of calm and excitement.

Zhang et al. [ZLR16] developed dependency-sensitive convolutional neural networks, which can capture long-term dependencies without relying on external parsers as in recursive neural networks. The model consists of a convolutional layer on top of LSTM layer. When the input is a sentence, the LSTM layer sequentially processes the embedding vectors of words to capture long-distance dependency within the sentence. The convolutional layer extracts the task-specific features from the LSTM layer for sentiment classification. When the input is a document, a document-level LSTM layer is added above the first LSTM layer to capture the linguistic relationships between sentences. The convolutional layer is set on top of the second LSTM layer as before. The architecture of for the document modeling is very similar to the previous work [TQL15a], except that the average pooling layer is replaced by a convolutional layer. Zhou et al. [ZQZ$^+$16] also studied the function of the pooling layer in sentiment classification. They replaced a 1D-max-over-time pooling layer with a 2D pooling layer on top of a bi-directional LSTM layer.

Hsu et al. [HMJS17] developed a hybrid CNN-RNN framework for sentence classification, in which they explicitly modeled the relationships between phrases and word sequences in each sentence. In fact, the phrases are represented by the outputs of the convolutional layer with windows of different widths. A GRU layer with attention mechanism over these phrases provides additional features to another GRU at higher level, which predicts the class label of the given sentence.

Embedding Methods

Word embedding is essential for neural networks in NLP. A pre-trained word embedding dictionary is useful because of its great capability of capturing the semantics of words from large scale unlabeled datasets. Word embedding at sentence level can be used for polarity classification directly with other off-shelf classifiers. Word embedding can also boost the performance of sentiment analysis models.

Alghunaim et al. [AMCG15] investigated the effectiveness of word vector representations for three subtasks in aspect based sentiment analysis: aspect term extraction, aspect category detection, and aspect sentiment prediction. They found that the performance of classic models, such as CRF and SVM, is improved on ABSA tasks with vector-based features.

Li et al. [LZL$^+$16] introduced both n-grams and weighting techniques into neural bag-of-words models, which can be regarded as a neural or distributed baseline like naive Bayesian SVM. Most of the neural models learn embeddings only for individual words. The authors showed that learning such embedding vectors for bi-gram can further enrich the semantics of text representations. They provided strong baselines on a range of text classification tasks, including sentiment classification.

Zhang et al. [ZRW16] utilized multiple groups of word embeddings, applied CNNs independently to each group, then concatenated the generated feature vectors of 1D-max-pooling layers for the classification layer. The authors also exploited different regularization penalties on network weights.

Paragraph Vector [LM14] is an unsupervised algorithm that learns fixed-length embedding vectors from variable-length pieces of text. Based on the word2vec model [MCCD13], the document-level dense embedding is trained to predict all the words in the input document. Skip-thought [KZS$^+$15] is another unsupervised model. Inspired by skip-gram model [MSC$^+$13], the author proposed a sentence-level embedding method, which is similar to RNN encoder-decoder framework in neural machine translation. An encoder compresses the current sentence into a vector, based on which two decoders generate the previous and the next sentence respectively. Yu et al. [YWLZ17] proposed a post-processing method to refine existing semantically oriented word vectors using sentiment lexicons.

3.5.2 Neural Networks for Sentiment Polarity Classification

This section surveys neural networks specially designed for sentiment polarity classification, such as numerical rating prediction and user and product modeling. We summarize the relevant work in the follow Table 3.5.2.

Table 3.2: The neural networks for sentiment classification

Name	Datasets	Base Model	Highlights
Tang et al. [TQLY15]	Yelp, RT05		user modeling
Tang et al. [TQL15b]	IMDB, Yelp	CNN	user and product modeling
Chen et al. [CST$^+$16]	IMDB, Yelp	LSTM	user and product modeling using attention mechanism
Chen et al. [CZLZ16]	Yelp, Dianping	CNN	vector representations of users and tiems
Guan et al. [GCZ$^+$16]	Amazon	CNN	weakly-supervised

Tang et al. [TQLY15] considered that different reviewers would have different sentiment strengths toward same words. They proposed a neural network for document-level rating prediction task by taking user information into account. Each word is represented as a vector and each user as a matrix. The word embeddings can be modified by a user preference matrix which maps the original word vector to the user-specific representations. The modified word representations are recursively composed by the tanh function to produce document-level vectors, which are used for rating prediction. Later, the authors modeled user and product information together in continuous vector spaces for document-level sentiment classification task [TQL15b].

Chen et al. [CST+16] studied two problems in the previous work [TQLY15]: there is not enough review data for training the preference matrix for each user, and the characteristics of users should be generated on the semantic level. They proposed a two-layer LSTM for document-level sentiment classification with attention mechanisms, in which user and product vectors are used as query vectors to attend on important semantic information at word level and sentence level.

Chen et al. [CZLZ16] used paragraph2vec to learn the distributed representations of users and items. The user vector is shared across the comments written by the same user, and the item vector is shared across the comments written on the same item. The concatenated vector of the user vector, the item vector and the context vector of the surrounding words is used to predict the target word. The learned item vector and the user vector is used to predict the sentiment score on unobserved user-item pairs without texts.

Guan et al. [GCZ+16] proposed a neural network for sentence sentiment classification. First, CNN [Kim14] with a ranking loss is used to learn an embedding space which reflects the general sentiment distribution of sentences, so that sentences with same weak labels to be close to each other. Then a classification layer is applied on top of the embedding layer with sentence labels to fine tune the network.

3.6 Extraction and Polarity Classification for Aspect Terms

Most reviewers comment about products or services on multiple aspects or features of the target. Such fine-grained opinion information can facilitate more insightful summarization and opinion retrieval. Opinion-oriented extraction [PL08] is the task to extract continuous text spans that discuss aspects of targets or opinions that associated with these aspects. It could be formulated as standard named entity recognition. Hence, it is not surprising that techniques such as condition random field [LMP01, RR09] can be immediately applied.

Recent deep learning methods for named entity recognition combine recurrent neural networks with conditional random filed. Chiu et al. [CN15] presented a hybrid model of bi-directional LSTMs and CNNs that learns both character- and word-level features. The features extracted by CNN from characters of each word are concatenated with word embeddings and additional word features. The result vectors are fed to the bi-directional LSTMs and then to the output layers that predict the scores of tags encoded in BIO-scheme. Lample et al. [LBS$^+$16] used a similar architecture as in [CN15], but they chose a bi-directional LSTMs to learn character-level features. Ma et al. [MH16] also proposed a very similar model with a CRF-like loss function.

In the work of Irsoy et al. [IC14], a deep bidirectional recurrent neural network was proposed for opinion expression extraction, in which the problem is formulated as a token-level sequence-labeling task. Stacking multiple layers of RNNs makes themselves having more ability of semantic abstraction. Experiments show that deep, narrow RNNs outperform traditional shallow, wide RNNs with the same number of parameters.

Yin et al. [YWD$^+$16] took words and dependency paths into account, which are shown important in aspect term extraction. They learned distributed representations of words and the dependency paths between them in an embedding space. The long-range dependency paths are modeled by a RNN. The learned embedding features of words and their dependencies can be utilized as features in conditional random field for aspect term extraction.

The task of aspect term extraction could be handled with sentiment classification together in a multi-task learning model. Wang et al. [WPDX16] utilized the dependency parse tree and trained a joint model that integrates recursive neural networks and conditional random field for the extraction of aspect and opinion terms. The underlying dependency structure provides a way for related aspect and opinion terms to interact with each other.

Later, Wang et al. [WPDX17] attempted to solve the task of aspect and opinion terms co-extraction by coupled attention mechanisms. This model uses multiple layers to refine the query vectors, which are called prototypes in the paper, in attention mechanism. In each layer, the new query vector is generated from a score function with previous prototypes and a GRU. In the last layer, the outputs of all GRU cells are used to predict the labels of each words. They constructed the layers for aspect terms and opinion terms to have double propagation effect.

Li et al. [LGM17] considered the task of detecting targets and classifying polarity towards the identified entities into positive, negative, or neutral. Instead of decomposing into two separate tasks, the authors proposed an end-to-end multi-task neural network, namely AttNet, which is equipped with a shared memory module to allow two connected tasks to learn from each other.

Aspect-term polarity classification is a downstream subtask of aspect-term extraction. The goal of the task is to classify the polarity of the identified term into positive or negative. For example, in a sentence of a product review, the reviewer likes one feature of the product, but dislikes other features. Nguyen et al. [NS15] extended recursive neural network to identify the entity sentiment by using syntactic information from both dependency and constituent trees of the sentence.

Zhang et al. [ZZV16] used gated neural network to model the syntax and semantics of the target, and the interaction between the surrounding contexts and the target. They explicitly modeled the interaction between the left context, the right context and the target. The outputs of GRUs are fed to three pooling layers and five gates to form the representation of the sentence. A softmax layer is used to predict the polarity of the target.

Similarly, Tang et al. [TQFL16] proposed a LSTM-based model which consists of two disconnected LSTM chains: one to model the preceding contexts, and the other to model the following contexts surrounding the entity. The outputs of both LSTMs can be used for sentiment classification. Afterwards, Tang et al. [TQL16] applied attention mechanism and explicit memory [SSWF15] for aspect-term polarity classification. Each layer is content- and location- based attention model, which captures the importance of each context word and then utilizes this information to compute continuous text representation. The output representation of the last layer is used for sentiment prediction.

Chen et al. [CSBY17] also adopted a similar strategy as in [TQL16]. The framework uses a bi-directional LSTM to produce the memory for the corresponding input word instead of directly using word embeddings. The memory slices are weighted according to their relative positions, so that different targets would have different memory features. Then multiple attentions are built on the weighted memory, the results of which are combined by GRUs to predict the sentiment on the target.

Ma et al. [MLZW17] modeled target terms and contexts simultaneously, and also considered that targets are also composed of many words. The representations of targets and contexts are determined by each other via two attention layers respectively. In the proposed network, the context representation is supervised by the attention layer associated with a target. Meanwhile, the interactive information from the context helps the modeling of the target. Finally, the vectors from the two attention layers are concatenated together to predict the sentiment of the targets. Yang et al. [YTW$^+$17] explored two types of attention mechanisms on LSTM: multiplication and concatenation.

3.7 Aspect Category Polarity Classification

Aspect category polarity classification is referred as the classification problem of the sentiment polarity about the given aspect associated with the input text. Lakkaraju et al. [LSM14] jointly modeled aspect extraction and sentiment analysis, which can capture subtle dependencies. They slightly changed recursive neural tensor network so that the top layer can output the class label as well as the sentiment class.

Wang et al. [WHZZ16] extended LSTM with attention mechanisms. The embedding vector of the given aspect are concatenated with the embedding of words to be the inputs of LSTM. Then an attention layer computes the final representation vectors by attending to the most important parts of the input sentence for sentiment prediction. Evaluations on SemEval 2014 indicated better accuracy then previous work [TQL16].

Ruder et al. [RGB16a] considered the inter-dependencies of sentences in a review and built a hierarchical bidirectional LSTM model, which can leverage both intra- and inter-sentence relations. Specifically, the sentence-level LSTMs output the representations of sentences to the review-level LSTMs with the embedding vector of the given aspect. The outputs of the upper level LSTMs are used for the sentiment prediction of input documents.

Yin et al. [YSZ17b] formulated the document-level aspect sentiment classification as machine comprehension problem. The hierarchical architecture builds difference representations at both word and sentence levels interacting with aspect questions.

3.8 Review Summarization and Generation

Summarization is an active research topic in NLP. Many researchers begin to focus on neural network based models for abstractive summarization, which attempts to produce a bottom-up summary [RCW15]. Inspired by neural machine translation, Rush et al. [RCW15] explored a fully data-driven approach which has less linguistic structure. This section surveys research work about summarization on review data.

Tang et al. [TYC$^+$16] applied natural language generation on the problem of review generation. They adopted encoder-decoder framework, in which a simple one-layer encoder encodes semantic representation from some contexts and a LSTM decoder generates word sequences depending on the semantic representation. To propagate the encoded information, a gating mechanism is applied on the decoder to control when the encoded representation from the contexts is accessed.

Wang et al. [WL16] also used the encoder and decoder framework, but enhanced it by attention mechanism to perform the summarization over a set of text units. To digest a set of reviews, they feed only the important text determined by an importance score from a linear regression. The score is measured by the number of overlapping words between each text unit and the gold standard summary.

Dong et al. [DHW$^+$17] proposed an attention-enhanced attribute-to-sequence model to generate product reviews for any given attribute information. The attribute encoder which is based on multi-layer perceptrons encodes input attributes into vector representations that are used as latent factors for generating reviews. The decoder which is based on multiple layers of RNN generates review text according to the encoded vectors. The attention mechanism learns soft alignments between generated words and attributes.

The researchers from OpenAI [RJS17] explored unsupervised learning in the task of learning distributed representations of sentences. They collected over 82 million product reviews as the training data set, and trained a large single layer multiplicative LSTM [KLMR16] at character-level for one month. A logistic regression classifier trained on the representation generated by LSTM obtains the-state-of-the-art accuracy for sentiment classification. They also discovered a single unit in the inner cell vector of LSTM that directly corresponds to sentiment.

3.9 Conclusion

In this chapter, we summarize the recent advances for deep learning models on review data. We briefly review the history of neural networks and deep learning, then focus on basic components for today's neural models: convolutional neural network, recurrent neural network, word embedding methods, attention mechanisms, and memory networks. We narrow down to some interesting tasks related to review data: sentiment polarity classification at many levels, aspect term extraction, and automatic summarization. Sentiment polarity classification on reviews is to identify the reviewer's sentiment attitude expressed in the given level. There are sentence level, aspect . At review level, the target entity could be the whole object like a hotel. At the aspect level, the entity could be service, value. At sentence level, the target could be a specific real object like a bed, a specific dish. Aspect term extraction proceeds the sentiment analysis. It detects the interesting entities at sentence level and passes them to other downstream tasks. Review summarization gives an abstract of given reviews, which would provide more information in words rather than just the counts of binary sentiment scores. Compared with traditional methods, deep learning could save a lot of amount of work on feature engineering and can provide end-to-end training framework.

CHAPTER 4

MTNA: A Neural Multi-task Model for Aspect Category Classification and Aspect Term Extraction On Restaurant Reviews

4.1 Introduction

ABSA [LZ12, PGP$^+$16] is defined to extract fine-grained insights such as named entities, aspects, and sentiment polarities, which facilitate users to digest reviews without reading. Although there is an abundant existing work of sentiment analysis, people often ignore other two fundamental tasks in ABSA: aspect category classification (ACC) and aspect term extraction (ATE).

Given a predefined set of aspect categories, aspect category classification aims to identify all the aspects discussed in a sentence. aspect term extraction is to recognize the word tokens of target entities. For example in restaurant reviews, suppose we have two aspects `PRICE` and `FOOD`. In the sentence "*The fish is carefully selected from all over the world and taste fresh and delicious.*", the aspect category is `FOOD`, the aspect term is `fish`. There could be multiple aspect categories implied in one single sentence; while in other sentences, there might be even no word token corresponding to the aspect category due to noisy aspect labels or fuzzy aspect definition. For example, the sentence "*I had a great experience.*" expresses positive attitude towards the aspect `RESTAURANT`, but there is no corresponding aspect term.

Lots of previous work has been proposed towards these two subtasks separately. Aspect category classification is often viewed as a supervised classification task. Off-shelf methods, such as logistic regression, support vector machines, and neural networks [TS16] can be immediately implemented. On the other hand, aspect term extraction is usually formulated as a sequence labeling problem. Traditional methods for this task include Conditional Random Field (CRF) and Hidden Markov Model (HMM) [RR09, LJM15]. Most of the existing work heavily relies on hand-crafted features, such as bi-gram, Part-of-Speech (POS) tags, word prefix, and word suffix.

Recognizing the commonalities between ACC and ATE task can boost the performance of both of them. We define aspect categories as a set of prefixed labels; while aspect terms are the informative words appear in the given sentence. There is a one-to-many relationship between aspect categories and aspect terms. For example, the aspect `FOOD` should cover all the food-related entities, such as `sushi`, `chicken`, and so on. On the other side, the occurrence of anything about food indicates a high probability that the sentence is about the aspect `FOOD`. Therefore, the aspect information of a whole sentence can make it easier to differentiate the aspect terms from unrelated words; while the recognized aspect terms could be used as the hints for predicting the aspect categories.

Neural networks have gained tremendous popularity and success in text classification [Kim14], machine translation [SVL14, BCB14], and text summarization [RCW15] due to the representational power of deep neural networks and the effective attention mechanism. The application of recurrent neural networks [IC14, LJM15] and convolutional neural networks [TS16] on ABSA has demonstrated the superior performance compared with traditional methods.

In this chapter, we consider aspect category classification and aspect term extraction together under a multi-tasking setting. Multi-task learning has been studied in deep learning system [CWB$^+$11], a unified neural network , where various basic natural language processing tasks are improved significantly when trained jointly. We conduct extensive experiments and analysis on SemEval-2016 dataset. Our model outperforms the conventional methods and competing deep learning models that tackle two problems separately.

The remaining part of this chapter is organized as follows. Section 4.2 briefly introduces the previous work on both tasks. Section 4.4 describes our multi-tasking model in a bottom-up way, and how the layers of each task learn useful information from each other. Section 4.5

and Section 4.5.4 present the comparative study of models on SemEval-2016 Task 5 dataset. Section 4.6 summarizes this chapter and proposes future work.

4.2 Related Work

Aspect based sentiment analysis on review data is a hot research topic [LZ12], which includes a series of correlated subtasks. There is much existing work for both target expression detection and aspect category classification.

Aspect Term Extraction. Aspect term extraction is usually modeled as a sequence labeling problem, which is related to named entity recognition task in natural language processing. Hu et al. [HL04] use association mining and WordNet to identify opinion sentences. Linear conditional random field is one of the well-known methods for named entity recognition in natural language processing. It can be immediately applied to the target expression detection problem [YC13]. However, traditional methods for this task including Conditional Random Field (CRF) and Hidden Markov Model (HMM) [RR09, LJM15] heavily relies on hand-crafted features, such as bi-gram, Part-of-Speech (POS) tags, word prefix, and word suffix.

Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM) [HS97] have been successfully applied to target detection [IC14, LJM15]. Usually, a bi-directional LSTM layer is built on a set of embedding layers encoding word features, POS tag features, and other linguistic features. The above softmax layer predicts the corresponding tag labels, such as BIO tagging scheme. Since it can be easily modeled as a special case of named entity recognition, more complicated methods for named entity recognition [CN15, MH16, LBS$^+$16] can be implemented. For example, character-level convolutional neural network could be added as an additional input layer [CN15]. Conditional random field layer could replace the top softmax layer to make the tagging results more coherent [MH16]. In the workshop of SemEval-2016, recurrent neural network is used to extract features for the subsequent CRF prediction [TS16].

Aspect Category Classification. The methods for analyzing aspect categories of reviews can be categorized into unsupervised and supervised methods.

Topic models model reviews with a mixture of latent topic variables and sentiment variables [TM08a, ME12]. Each topic is a probability distribution over the vocabulary words. The generated topics could be considered as aspects. However it needs further post-processing to align the generated topics with the predefined aspects. More often than not, the aspect classification task is treated as a multi-label classification problem. The one-vs-all strategy is used to train a binary classifier for each aspect. Off-the-shelf classifiers such as support vector machine, logistic regression with various features such as n-grams, tf-idf, word embedding have achieved satisfactory results [TS16].

Convolutional Neural Networks (CNNs) [LBBH98] have been proposed for sentiment classification [CWB$^+$11, Kim14, KGB14]. Collobert et al. [CWB$^+$11] proposed a multi-task learning system using deep learning methods for various natural language processing tasks. However, the system with window approach cannot be jointly trained with that using sentence window approach. Moreover, only embedding layer (lookup table) and linear layer are shared among tasks, which limits the utilization of shared information. Kim et al. [Kim14] used a shallow and wide structure in CNN, which is still a top method on text classification task.

To our best knowledge, our model is the first work that model ACC and ATE tasks in an integrated way. Most relevant work is Dependency Sensitive Convolutional Neural Networks (DSCNN) [ZLR16]. However, the goal of DSCNN is just for text classification. LSTM is used as feature extractor to capture the long term dependencies, which is difficult for convolution layer to learn. Our model is designed for multi-task learning on review data.

4.3 Problem Definition

In this section, we specifically define two ABSA tasks: aspect category classification (ACC) and aspect term extraction (ATE), then present an end-to-end model MTNA (Multi-Task neural Networks for Aspect classification and extraction) that jointly solve the two tasks.

We define ACC as a supervised classification task where the sentence should be labeled according to a subset of predefined aspect labels, and ATE as a sequential labeling task where the word tokens related to the given aspects should be tagged according to a predefined tagging scheme, such as IOB (Inside, Outside, Beginning). Concretely, we have a set of aspect labels $\{A_1, A_2, \ldots, A_k\}$. Given a sentence $[x_1, x_2, \ldots, x_T]$, a model is required to choose a subset of aspect labels for the sentence; meanwhile it should mark aspect terms according to a tagging scheme, such as the BIO scheme in named entity recognition task.

4.4 The Multi-task Learning Model

We assume the words x_i in the given text are indexed in a vocabulary $\mathbf{V}$, e.g., $x_i \in \{1, 2, \ldots, V\}$. The word embedding layer transforms the word indices to a real valued vector $\mathbf{x}_i \in \mathbb{R}^d$ with a pre-trained word embedding matrix [MSC$^+$13, PSM14]. d is the dimension size of embedding vectors. T is the length of the sentence. Each sentence is then represented by a matrix $\mathbf{S} \in \mathbb{R}^{T \times d} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T]$.

First, we describe how to apply LSTM for aspect term extraction. RNN [RHW86] is a family of networks which apply a recursive composition function on sequential data. RNN sequentially composes inputs with Equation 4.1. At each time stamp t, it takes an affine transformation of the current input $\mathbf{x}_t$ and the previous hidden state $\mathbf{h}_{t-1}$, then apply an activation function σ to get the current hidden state $\mathbf{h}_t$. Specifically,

$$\mathbf{h}_t = \sigma(\mathbf{W}^{(r)}\mathbf{x}_t + \mathbf{U}\mathbf{h}_{t-1} + \mathbf{b}) \quad , \tag{4.1}$$

where $\mathbf{W}$, $\mathbf{U}$, and $\mathbf{b}$ are the parameters transforming the input $\mathbf{x}$ and the hidden state $\mathbf{h}_{t-1}$ respectively. LSTM [HS97] is designed to overcome the exploding and vanishing gradient problem of RNN by introducing memory cells and a group of adaptive nonlinear gates which control the information flow of the network. Formally, LSTM uses the input vector $\mathbf{x}$ and the previous hidden vector $\mathbf{h}$ to compute three gates and one candidate memory cell: the input gate $\mathbf{i}_t$, the forget gate $\mathbf{f}_t$, the output gate $\mathbf{o}_t$, and the candidate $\mathbf{u}_t$. The final cell state $\mathbf{c}_t$ and the output vector $\mathbf{h}_t$ is then updated by

$$\mathbf{i}_t = \sigma(\mathbf{W}^{(i)}\mathbf{x}_t + \mathbf{U}^{(i)}\mathbf{h}_{t-1} + \mathbf{b}^{(i)}) \tag{4.2}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}^{(f)}\mathbf{x}_t + \mathbf{U}^{(f)}\mathbf{h}_{t-1} + \mathbf{b}^{(f)}) \tag{4.3}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}^{(o)}\mathbf{x}_t + \mathbf{U}^{(o)}\mathbf{h}_{t-1} + \mathbf{b}^{(o)}) \tag{4.4}$$

$$\mathbf{u}_t = \tanh(\mathbf{W}^{(u)}\mathbf{x}_t + \mathbf{U}^{(u)}\mathbf{h}_{t-1} + \mathbf{b}^{(u)}) \tag{4.5}$$

$$\mathbf{c}_t = \mathbf{i}_t \odot \mathbf{u}_t + \mathbf{f}_t \odot \mathbf{c}_{t-1} \tag{4.6}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \quad , \tag{4.7}$$

where σ denotes the sigmoid function, $\odot$ is element-wise multiplication, $\mathbf{W}, \mathbf{U}, \mathbf{b}$ are the matrix parameters of gates.

We apply a bi-directional LSTM for aspect term extraction, which could be taken as a special case of named entity recognition [CN15, LJM15, LBS$^+$16, MH16]. Bi-directional LSTM has two parallel LSTMs, in which one LSTM composes words forwards and the other one backwards, so that any cell can access the past and the future information. At any time stamp t, bi-directional LSTM takes word vector $\mathbf{x}_t$ and output concatenated hidden vectors $\mathbf{h}_t = [\mathbf{h}_{\text{forward}_t}, \mathbf{h}_{\text{backward}_t}]$. $\mathbf{h}_t$ is then fed to a fully-connected layer with softmax nonlinearity function to predict the probabilities of the tags of the word t. The loss function for ATE task is the sum of the cross-entropy loss for all words:

$$L_{\text{ate}} = -\frac{1}{NT}\sum_{i=1}^{N}\sum_{j=1}^{T}\sum_{k}\left[z_{ijk}\log \hat{z}_{ijk}\right] \ , \tag{4.8}$$

where $\hat{z}_{ijk}$ is the probability of the tag k for the word x_j in the i-th sentence, and z_{ijk} is the groundtruth.

Second, we apply CNN model for text classification [Kim14] on ACC task. CNN uses a one-dimensional convolutional layer to compute features between its inputs and a kernel $\mathbf{W} \in \mathbb{R}^{d \times w}$ over all possible windows of the inputs, where w is the width of the kernel. One convolution kernel $\mathbf{W}$ generates a feature $\mathbf{k} \in \mathbb{R}^{(T-w+1)}$ on an input sentence, if we do not pad zero vectors around the input.

$$k_i = f(\mathbf{W}^{(c)} \odot [\mathbf{x}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_{i+k-1}] + b) \ , \tag{4.9}$$

where b is the bias, and f is a nonlinearity active function. There could be many convolutional kernels with different kernel sizes. 1D max-over-time pooling [CWB+11] is then applied over each feature map to extract the maximum value, $c_i = \max_i k_i$. In practice, the max-pooling layer outputs a large concatenated feature vector $\mathbf{c} \in \mathbb{R}^{d'}$, where d' is the number of convolutional kernels. Finally, a fully connected output layer with softmax function to predict the probabilities of aspect categories. The loss function for aspect category classification is binary cross entropy loss.

$$L_{\text{acc}} = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log \hat{y}_i + (1 - y_i)(\log(1 - \hat{y}_i))] \ , \tag{4.10}$$

where $\hat{y}_i$ is the predicted probability of an aspect label for the i-th sentence.

Now, we are ready to build a multi-task learning model for ACC task and ATE task. It should be noted that ACC task and ATE task are closely related. Aspect terms often implies the related aspect category. If the names of dishes appear in a sentence, it is easy to infer that this sentence is about the aspect FOOD and vice-versa. Multi-task learning can help the model of each task to focus its attention to relevant features, when the other task supports these features with evidence [Rud17]. Moreover, multi-task learning can obtain a common representation for all the tasks in the shared layers, which reduces noise in each task [Rud17]. We combine bi-directional LSTM for ATE and CNN for ACC together in a

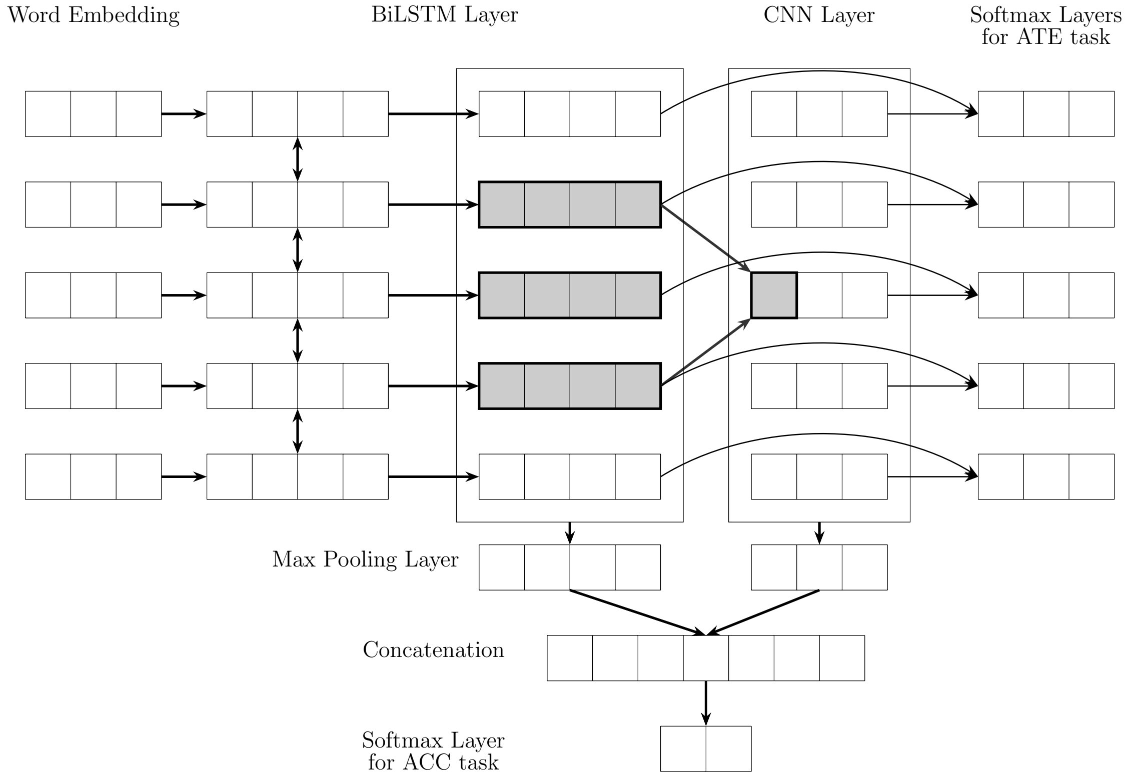


Figure 4.1: MTNA on a sequence of five words. The multi-task learning neural network combines BiLSTM and CNN layers together for ATE and ACC task respectively. One convolutional operation on BiLSTM layer is shown in the graph.

multi-task framework. The convolutional layers for ACC task can utilize extra information learned in ATE task so that the convolutional layers can focus on informative features. The tag prediction at each word in ATE task can also receive the distilled n-gram features of the surrounding words via convolutional operations.

The architecture of our model is shown in Figure 4.1. Specifically, A bi-directional LSTM is applied on the outputs of word embedding layer $\mathbf{S}$. The feature $\mathbf{h}_t$ is processed by a one-dimensional convolution layer with many kernels, so that the new feature $\mathbf{c}_t$ incorporates the information of words that are in the receptive field of the convolutions. For ATE task, we use one softmax layer for each word in the given sentence to predict its tag. We further add skip connections from the bi-directional LSTM layers to the softmax layers [HZRS16a]. Therefore, the inputs to the softmax layers for ATE task are:

$$\mathbf{h}'_t = [\mathbf{h}_{\text{forward}_t}, \mathbf{h}_{\text{backward}_t}, k_i^{(1)}, \ldots, k_i^{(n)}] \,, \tag{4.11}$$

where n is the number of convolution kernels. For ACC task, we use a 1D max-over-time pooling layer on the bi-directional LSTM and convolutional layer. The concatenated outputs of the pooling layer is fed to a softmax layer to predict the probabilities of aspect categories for the sentence. The inputs to the softmax layer for ACC task are:

$$\mathbf{c} = [\max\{\mathbf{h}_1, \ldots, \mathbf{h}_T\}, c^{(1)}, \ldots, c^{(n)}] \tag{4.12}$$

The final loss function of our model is a weighted sum of the loss functions of ACC task and ATE task. $L = L_{\text{acc}} + \lambda L_{\text{ate}}$, where λ is the weight parameter. L_{acc} is the cross-entropy loss function for ACC task; L_{ate} is the sentence-level log-likelihood for ATE task [CWB$^+$11, LBS$^+$16].

4.5 Experiments

4.5.1 Datasets

The data set in our experiments is the sentences associated with pairs of word tags and aspect labels as follows,

```
<text>Service was divine, oysters where
a sensual as they come,
and the price can’t be beat!</text>
    <Opinions>
        <Opinion target="Service"
         category="SERVICE"/>
        <Opinion target="oysters"
```

```
     category="FOOD"/>
    <Opinion target="NULL"
     category="RESTAURANT"/>
</Opinions> .
```

In the Opinion tag, the target attribute is the word tag in the target expression detection task and the category attribute is the aspect category label in the aspect classification task.

We consider three data sets from SemEval workshops in recent years: SemEval 2014 Task 4 (SE14) [PGP$^+$14], SemEval 2015 Task 12 (SE15) [PGP$^+$15], and SemEval 2016 Task 5 (SE16) [PGP$^+$16]. We use the reviews in restaurant domain for all of them, and process SE14 into the same data format as the others. Each data set contains 2000 - 3000 sentences. For SE15 and SE16, an aspect label is a combination of an aspect and an attribute, like “Food#Price”. There are 6 main aspects and total 12 configurations in SE15, SE16, while 5 aspects in SE14.

In SE16 dataset, the training data set consists of 2,000 sentences and 2,507 tuples. The test data set contains 676 sentences and 859 tuples. The dataset contains six aspect categories: FOOD, RESTAURANT, SERVICE, AMBIENCE, DRINKS, LOCATION. However the number of sentences associated with DRINKS and LOCATION is much less than those associated with other four aspects. The statistics are shown in Table 4.1. It should be noticed that not all of Opinion contain target attribute, especially for the aspect RESTAURANT, because the concept is not as clear as other aspects. Even the names of restaurants are labeled as aspect terms, when the whole sentence labeled as RESTAURANT. There is no explicit opinion term in some sentences.

4.5.2 Experiment Setup

Following the experiment settings used by most competitors [TS16, KEB16, Mac16] in SemEval 2016, we convert the multi-label aspect classification into multiple one-vs-all

Aspect	# of sentences	
	Training	Test
FOOD	891 (670)	296 (230)
RESTAURANT	598 (245)	196 (81)
SERVICE	419 (297)	145 (100)
AMBIENCE	226 (202)	57 (51)

Table 4.1: The statistics of SemEval-2016 Restaurant Review Dataset. The numbers in parenthesis are the number of sentences which are associated at least one target.

binary classifications. F1-score is used to measure the performance of each model for ACC task, and another F1 measure adapted for ATE task. We set the tagging scheme for ATE task. Each tag can have different types, such as `I-FOOD`, `B-FOOD`, `I-SERVICE`, and `B-SERVICE`. In each multi-task learning model, words are tagged with IOB labels that is only related to the aspect label. For example, if the aspect label is `FOOD`, the possible tag could be one of `I-FOOD`, `B-FOOD`, and `O`.

For MTNA model, we use the pre-trained word embedding GloVe [PSM14] of 200 dimensions to initialize the embedding layer. The word vectors that are out of GloVe vocabulary are randomly initialized between -0.1 and 0.1. During the training process, the embedding vectors are fine-tuned. We choose three kinds of convolution kernels which have the width of 3, 4, 5. Each of them has 100 kernels [Kim14]. We use tanh function as the nonlinear active function in convolution layers based on the results of cross validation. We train the model with Adadelta [Zei12]. For each multi-task learning model, a 5-fold cross validations is used to tune other hyper-parameters: mini-batch size from $\{10, 20, 50\}$, dropout rate from $\{0.1, 0.2, 0.5\}$, the dimension of LSTM cells from $\{100, 200, 500\}$, and the weight λ in the loss function from $\{0.1, 1, 10\}$.

4.5.3 Compared Methods

We compare models from two categories: off-shelf methods and neural networks for each task.

	SE14		SE15		SE16	
	ACC	ATE	ACC	ATE	ACC	ATE
Top models	88.57	**84.01**	62.68	67.11	73.03	72.34
BiLSTM-CRF	-	83.24	-	66.82	-	71.87
MTNA-s	87.95	-	64.32	-	75.69	-
MTNA	**88.91**	83.65	**65.97**	**67.73**	**76.42**	**72.95**

Table 4.2: Comparison results in F1 scores on three datasets.

Top models in SemEval. For ACC task, NRC-Can [KZCM14] and NLANGP [TS15] are top models in 2014 and 2015 respectively, both of which use SVM. NLANG [TS16] adopts CNN-like neural network in 2016. For ATE task, CRF [TW14, TS15, TS16] is the best model on all of three data sets.

CNN. CNN has been adopted in aspect category classification [TS16]. To assess whether TED task can help ACC, we build a CNN which has 1D convolution kernels of width 3,4,5. The number kernels for each different width is set 100. The embedding layers are also initialized with GloVe embeddings. 1D max-pooling layer follows convolution layers. The softmax layer outputs the label probability. The batch size and learning rate are tuned with 5-fold cross-validation as well.

BiLSTM-CRF. To assess whether CNN can improve the performance of ATE, we use a standard Bi-directional LSTM with CRF layer [LBS$^+$16] as the baseline to tag words.

MTNA-s. To evaluate to what extent that ATE loss function can improve the performance of the ACC task, we compare MTNA with its variance MTNA-s, the loss function of which does not include that of ATE task. However, this model keeps LSTM layer as a feature extractor before the convolution layers as MTNA does.

4.5.4 Results and Analysis

The comparison results of all methods on three datasets are shown in Table 4.2.

Model	Aspect Category Classification			AspectTerm Extraction		
	Food	Restaurant	Service	Food	Restaurant	Service
CNN	86.29	65.27	84.02	-	-	-
Bi-LSTM-CRF	-	-	-	73.96	54.34	87.55
MTNA-s	86.41	67.89	84.93	-	-	-
MTNA	87.33	66.07	86.09	74.67	56.59	88.70
	Ambience	Drinks	Location	Ambinece	Drinks	Location
CNN	81.55	67.36	69.25	-	-	-
Bi-LSTM-CRF	-	-	-	76.23	71.38	56.77
MTNA-s	81.08	69.23	70.06	-	-	-
MTNA	83.18	68.75	71.43	77.79	72.21	60.16

Table 4.3: F1 scores of models on SE16 across six aspects

On ACC task, MTNA outperforms over other compared methods, which are proposed for a single task and cannot utilize the information from the other task. On ATE task, there are small improvement compared with conditional random field. It empirically proves that multi-task learning can benefit both tasks. MTNA has higher F1-scores compared with BiLSTM-CRF. The results confirm the effectiveness of additional convolution features for the ATE task.

MTNA-s, a smaller model without layers for ATE task, also performs better than CNN. It proves that LSTM can provide the feature engineering which captures the long-distance dependency [ZLR16]. On the aspects other than RESTAURANT, MTNA-s has slightly lower scores than MTNA, which again demonstrates the effectiveness of multi-task learning.

To access the performance of methods across different aspects, we combine all sentences labeled by the same aspect regardless of any attribute, then conduct experiments as before. We re-implement CNN model, which is used in NLANG 2016. The results are as shown in Table 4.3. ACC task on the aspect RESTAURANT is more difficult than the task on other aspects. Both CNN and MTNA have lower F1-scores on this aspect. The reason is that some sentences have restaurant names as target terms. However, there are around 40.1% sentences with RESTAURANT label that do not have annotated words in the training dataset, 41.2% in test dataset. Meanwhile, all methods have better results in ATE task on the aspect SERVICE than on the other aspects, because target word tokens do not have much variety.

4.6 Conclusion

We introduce two important tasks, e.g., aspect category classification and aspect term extraction in aspect based sentiment analysis. We propose a multi-task learning model based on recurrent neural networks and convolutional neural networks to solve the two tasks at the same time. Finally, the comparative experiments demonstrate the effectiveness of our model across three public datasets. We can utilize other linguistic information, such as POS tags and the distributional representation learned from character level convolutional neural network in the future work.

CHAPTER 5

Gated Convolutional Neural Networks for Aspect Based Sentiment Analysis

5.1 Introduction

Opinion mining and sentiment analysis [PL08] on user-generated reviews can provide valuable information for providers and consumers. Instead of just predicting the overall sentiment polarity, fine-grained aspect based sentiment analysis (ABSA) [LZ12] is proposed to better understand reviews than traditional sentiment analysis. Generally, we are interested in the sentiment polarity toward the given aspect categories or target entities in the text, instead of the overall sentiment. A number of models have been developed for ABSA, but there are two different subtasks, namely aspect-category sentiment analysis (ACSA) and aspect-term sentiment analysis (ATSA). The goal of ACSA is to predict the sentiment polarity with regard to the given aspect, which is one of a few predefined categories. On the other hand, ATSA is to identify the sentiment polarity concerning the target entities that appear in the text instead, which could be a multi-word phrase or a single word. The number of distinct words contributing to aspect terms could be more than a thousand. For example, in the sentence "*Average to good Thai food, but terrible delivery.*", ATSA would ask the sentiment polarity towards the entity *Thai food*; while ACSA would ask the sentiment polarity toward the aspect *service*, even though the word *service* does not appear in the sentence.

Many existing models use LSTM layers [HS97] to distill sentiment information from embedding vectors, and apply attention mechanisms [BCB14] to enforce models to focus on the text spans related to the given aspect. Such models include Attention-based LSTM with Aspect Embedding (ATAE-LSTM) [WHZZ16] for ACSA; TD-LSTM (Target-Dependent Sentiment Classification) [TQFL16], Gated Neural Networks [ZZV16] and Recurrent Attention Memory Network [CSBY17] for ATSA. LSTMs sequentially process tokens in the given text. At each time stamp, the nonlinear gates control the information flows based on the previous status of hidden vectors and current inputs, where the sequential information is preserved. Because of the sequential nature of LSTMs, they can model long-range dependencies, but also take a lot of computing time during training. Attention mechanisms has been successfully used in many NLP tasks. In the work of ATAE-lSTM [WHZZ16], on the top of a LSTM layer, an attention layer first computes the alignment scores between context vectors and target vector; then carry out a weighted sum with the scores and the context vectors. However, the context vectors have to encode both the aspect and sentiment information, and the alignment scores are applied across all feature dimensions regardless of the differences across them. Both LSTM and attention layer are very time-consuming during training. LSTM has to process one token at a time. Attention layer involves exponential operation and aggregation of all alignment scores of all the words in the sentence [WHZZ16].

On the other hand, in models using CNN for sentiment classification [Kim14], the convolutional layer can efficiently extract n-gram features from the underlying word embedding layer, while the max-pooling layer can filter out the noises and keep the most informative features for the prediction of the final fully connected layer. Aspect based sentiment analysis further requires the model to be selective on the sentiment features. Therefore, how to make the model responds to the given aspect information while keeping low noise is critical to the task.

In this chapter, we propose a fast but effective architecture for ACSA and ATSA based on convolutions and gating mechanisms.

For ACSA subtask, we have two separate convolutional layers on the top of the embedding layer, whose outputs are combined by novel gating units. Convolutional layers with multiple filters can efficiently extract n-gram features at many granularities on each receptive field. The proposed gating units have two nonlinear gates, each of which is connected to a convolutional layer. With the given aspect information, they can selectively extract aspect-related sentiment information for the sentiment prediction. For example, in the sentence "*Average to good Thai food, but terrible delivery.*", when the aspect *food* is provided, the gating units automatically ignore the negative sentiment of aspect *delivery* from the second clause, and only output the positive sentiment from the first clause. Because each component of the proposed model can be easily parallelized, it has much less training time than the models based on LSTM and attention mechanisms. For ATSA subtask, where the aspect terms consist of multiple words, we extend our model to include another convolutional layer for the target expressions. We evaluate our models on the SemEval datasets, which contains restaurants and laptops reviews.

5.2 Related Work

In the area of recommendation system and data mining, matrix factorization, probabilistic graphical models often reply on intensive feature engineering to extract relevant opinion terms in the preprocessing step. The external parsing algorithms are often unreliable and inefficient when handling noisy user-generated text. For example, recommendation systems [ZLZ$^+$14, HCKC15] use grammatical and morphological analysis tools; SLUM [BLT17] requires Double Propagation [QLBC11] to extract opinion targets from text using Part-of-Speech tags and a set of fixed rules. Probabilistic graphical models [TM08a, ZJYL10, WLZ10, DQW$^+$14] are often based on bag-of-words assumption and work at document level, which suffers sparsity problem.

5.2.1 Neural Networks

Recently, neural networks have gained much popularity on sentiment analysis or text classification task. Tree-based recursive neural networks(RecNN) such as Recursive Neural Tensor Network [SPWC13], Tree-LSTM [TSM15], and Dynamic RecNN [LQH17], make use of syntactic interpretation of the sentence structure, but these methods suffer from time inefficiency and high parsing error on review text. Competitive results are achieved with simpler methods like recurrent neural network [LLJH15], or Deep Averaging Network [IMBGDI15]. Recurrent Neural Networks (RNNs) such as LSTM [HS97] and GRU [CGCB14], have been used for sentiment analysis on data instances having variable length [TQL15a, XCQH16, LXLZ15]. There is also a large body of research in NLP using convolutional neural networks (CNNs) [CWB$^+$11, KGB14, Kim14, CSBL16], which prove that convolution operations can capture compositional structure of texts with rich semantic information without laborious feature engineering.

5.2.2 Aspect based Sentiment Analysis

There is abundant research work on sentiment analysis with the name of "aspect based". It has been used to describe two different subtasks in the literature. We classify the existing work involving aspect extension into two main categories based on the descriptions of four sentiment analysis tasks in SemEval 2014 Task 4 [PGP$^+$14]: **Aspect-Term Sentiment Analysis** and **Aspect-Category Sentiment Analysis**.

Aspect-Term Sentiment Analysis. Most work falls into the first category, in which sentiment analysis is performed toward the labeled aspect words in the given sentence. A large body of literature tries to utilize the relation or position between the target words and the surrounding context words either by using the tree structure of dependency or by simply counting the number of words between them as a distance.

Recursive neural networks [LSM14, DWT$^+$14, WPDX16] rely on external syntactic parsers, which could be inaccurate and slow on noisy texts like tweets and reviews and may result in inferior performance.

Recurrent neural networks are building blocks for TD-LSTM [TQFL16, ZZV16]. The linear structure of a LSTM layer is truncated into three segments: the target words, the left context, and the right context. A fully-connected layer with gating units uses the outputs of LSTM layers to predict the sentiment polarity. Interactive attention networks [MLZW17] which is also based on LSTMs, use two attention layers to make context and target supervise the representation modeling of each other.

Memory network [WCB14] coupled with multiple-hop attention attempts to explicitly focus only on the most informative context area to infer the sentiment polarity towards the target word [TQL16, CSBY17]. Nonetheless, memory network simply bases its knowledge bank on the embedding vectors of individual words [TQL16], which makes itself hard to learn the opinion word enclosed in more complicated contexts. The problem is eased by using LSTMs as feature extractors and GRU in attention layers [CSBY17], but it needs unreliable word distance between surrounding words and target words to adjust and produce a unique memory for each target.

Aspect-Category Sentiment Analysis. In this category, the model is asked to predict the sentiment polarity toward a predefined aspect category, which may not appear in the input text. Attention-based LSTM with Aspect Embedding [WHZZ16] uses the aspect information to selectively attend on the representations generated by LSTMs. At the document level, Ruder et al. [RGB16a] considered the inter-dependencies of sentences in a review and built a hierarchical bidirectional LSTM model, in which the aspect embeddings are connected to the upper level LSTM that are used for the sentiment prediction of given documents. Yin et al. [YSZ17b] formulated the document-level aspect sentiment classification as machine comprehension problem by constructing pseudo question-answer pairs.

5.3 The Formal Definition of ACSA and ATSA

In this section, we define ACSA and ATSA tasks we are going to solve. We first focus on the task of ACSA problem. Given a sentence s made of a sequence of words $\{w_1, w_2, \dots, w_L\}$ and a predefined aspect a, the model is asked to predict the sentiment polarity y towards the aspect. The sentiment y could be one of positive, negative, or neutral. The aspect is represented by a word w_a but not necessarily appear in the sentence. It could be expressed in a rather implicit way. For example, in a simple sentence of a restaurant review "*I had a great experience.*", there is no aspect term, but it apparently comment on the aspect "*restaurant*".

On the contrary, in ATSA, aspect terms $w_i, w_{i+1}, \dots, w_{i+k}$ must appear in the input sentences. The sentiment polarity is defined the same as in ACSA. It might be easy to pinpoint the sentiment indicators in the surrounding contexts. For example, when the aspect term "*food*" is labeled in a long sentence, the model could just focus on several adjective words before or after the targets.

5.4 Gated Convolutional Network with Aspect Embedding on ACSA

In this section, we present a new model for ACSA and ATSA, namely Gated Convolutional network with Aspect Embedding (GCAE), which is more efficient and simpler than the recurrent network based models [WHZZ16, TQFL16, MLZW17, CSBY17]. Recurrent neural networks sequentially compose hidden vectors $\mathbf{h}_i = f(\mathbf{h}_{i-1}, w_i)$, which does not enable parallelization over inputs. In the attention layer, softmax normalization also has to wait for all the alignment scores computed by a similarity function. Hence, they cannot take advantage of highly-parallelized modern hardware and libraries. Our model is built on convolutional layers and gating units. Each convolutional filter computes n-gram features at different granularities from the embedding vectors at each position individually. The gating units on top of the convolutional layers at each position are also independent from each other. Therefore, our model is more suitable to parallel computing. Moreover, our model is equipped with two kinds of effective filtering mechanisms: the gating units on top of the convolutional layers and the max pooling layer, both of which can accurately generate and select aspect-related sentiment features.

We first briefly review the vanilla CNN for text classification [Kim14]. The model achieves state-of-the-art performance on many sentiment classification datasets [LCD17]. To the best of our knowledge, no CNN based model has been proposed for aspect based sentiment analysis.

The CNN model consists of an embedding layer, a one-dimension convolutional layer and a max-pooling layer. The embedding layer takes the indices $w_i \in \{1, 2, \ldots, V\}$ of input words and outputs the corresponding embedding vectors $\boldsymbol{v}_i \in \mathbb{R}^D$. D denotes the dimension size of the embedding vectors. V is the size of the vocabulary. Embedding vectors are initialized with pre-trained ones such as GloVe [PSM14], which would be fine-tuned during the training stage. The one-dimension convolutional layer convolves the inputs with multiple convolutional kernels of different widths. Each kernel corresponds a linguistic feature detector which extracts a specific pattern of n-gram at various granularities [KGB14]. Specifically, the input sentence is represented by a matrix through the embedding layer, $\mathbf{X} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_L]$. A convolutional filter $\mathbf{W}_c \in \mathbb{R}^{k,d}$ maps k words in the receptive field to a single feature. As we slide the filter across the whole sentence, we produce a sequence of new features $\mathbf{c} = [c_1, c_2, \ldots, c_L]$.

$$c_i = f(\mathbf{X}_{i:i+k} * \mathbf{W}_c + b_c) \quad , \tag{5.1}$$

where $b_c \in \mathbb{R}$ is the bias, f is a non-linear activation function such as tanh function, $*$ denotes convolution operation. If there are d_k filters of the same width, the output features form a matrix $\mathbf{C} \in \mathbb{R}^{d_k \times L_k}$. For each convolutional filter, the max-over-time pooling layer takes the maximal value among the generated convolutional features $\mathbf{c} \in \mathbb{R}^L$, resulting in

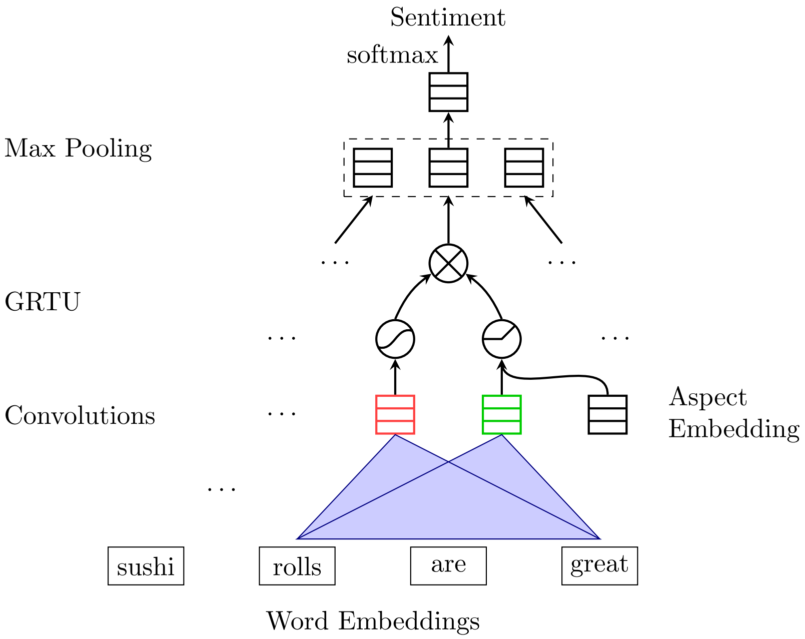


Figure 5.1: Illustration of our model GCAE for ACSA task. A pair of convolutional neuron computes features for a pair of gates: tanh gate and ReLU gate. The ReLU gate receives the given aspect information to control the propagation of sentiment features. The outputs of two gates are element-wisely multiplied for the max pooling layer.

a fixed-size vector whose size is equal to the number of filters d_k. Finally, a softmax layer uses the vector to predict the sentiment polarity of the input sentence.

Figure 5.1 illustrates the model architecture. The Gated Tanh-ReLU Units with aspect embedding (GTRU) are connected to two convolutional neurons at each position t. Specifically, we compute the features c_i as

$$a_i = \text{relu}(\mathbf{X}_{i:i+k} * \mathbf{W}_a + \mathbf{V}_a \boldsymbol{v}_a + b_a) \tag{5.2}$$

$$s_i = \tanh(\mathbf{X}_{i:i+k} * \mathbf{W}_s + b_s) \tag{5.3}$$

$$c_i = s_i \times a_i \quad , \tag{5.4}$$

where $\boldsymbol{v}_a$ is the embedding vector of the given aspect category in ACSA or computed by another CNN over aspect terms in ATSA. The two convolutions in Equation 5.2 and 5.3 are the same as the convolution in the vanilla CNN, but the convolutional features a_i includes the additional aspect information $\boldsymbol{v}_a$ with ReLU activation function. It means that s_i and

a_i are responsible for generating sentiment features and aspect features respectively. The above max-over-time pooling layer generates a fixed-size vector $\boldsymbol{e} \in \mathbb{R}^{d_k}$, which keeps the most salient sentiment features of the whole sentence. The final fully-connected layer with softmax function uses the vector $\boldsymbol{e}$ to predict the aspect sentiment $\hat{y}$. The model is trained by minimizing the cross-entropy loss between the ground-truth y and the predicted value $\hat{y}$ for all data samples.

$$\mathcal{L} = -\sum_i \sum_j y_i^j \log \hat{y}_i^j \quad , \tag{5.5}$$

where i is the index of the data sample, j is the index of the sentiment class.

5.5 Gating Mechanisms

The proposed Gated Tanh-ReLU Units control the path through which the sentiment information flows towards the pooling layer. The gating mechanisms have proven to be effective in LSTM. In aspect based sentiment analysis, it is very common that different aspects with different sentiments appear in one sentence. The ReLU gate in Equation 5.2 does not have upper bound on positive inputs but strictly zero on negative inputs. Therefore, it can output a similarity score according to the relevances between the given aspect information $\boldsymbol{v}_a$ and the aspect feature a_i at position t. If this score is zero, the sentiment features s_i would be blocked at the gate; otherwise, its magnitude would be likely to be amplified accordingly. The max-over-time pooling time would further remove the sentiment features which are not significant over the whole sentence.

In language modeling [DFAG17, KES$^+$16, vdOKE$^+$16, GAG$^+$17], Gated Tanh Units (GTU) and Gated Linear Units (GLU) have shown effectiveness of gating mechanisms. GTU is represented by $\tanh(\mathbf{X} * \mathbf{W} + b) \times \sigma(\mathbf{X} * \mathbf{V} + c)$, in which the sigmoid gates control features for predicting the next word in a stacked convolutional block. To overcome the gradient vanishing problem of GTU, GLU uses $(\mathbf{X} * \mathbf{W} + b) \times \sigma(\mathbf{X} * \mathbf{V} + c)$ instead, so that the gradients would not be downscaled to propagate through many stacked convolutional layers. However, a neural network that has only one convolutional layer would not suffer from gradient vanish problem during training. We show that on text classification problem, our GTRU is more effective than these two gating units.

5.6 Gated Convolutional Network with Aspect Embedding on ATSA

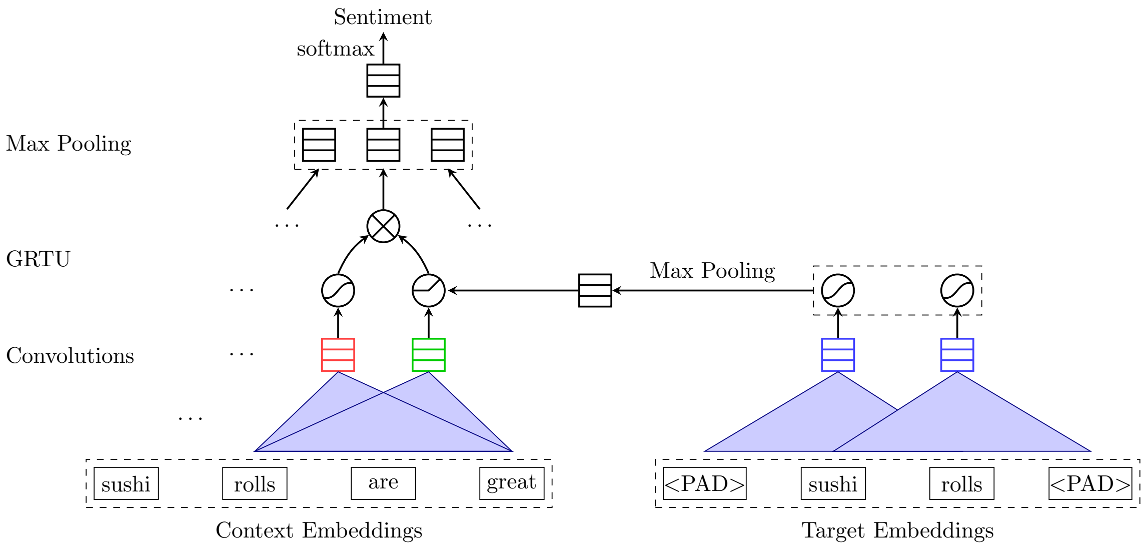


Figure 5.2: Illustration of model GCAE for ATSA task. It has an additional convolutional layer on aspect terms.

ATSA task is defined to predict the sentiment polarity of the aspect terms in the given sentence. We simply extend GCAE by adding a small convolutional layer on aspect terms, as shown in Figure 5.2. The aspect embeddings for the ReLU gate in GTRU are initialized with the pre-trained embeddings of the aspect words; while in GCAE, the aspect information to control the flow of sentiment features is provided by the outputs of the small CNN on aspect terms $[w_i, w_{i+1}, \dots, w_{i+k}]$. The additional CNN could extract the important features from the multiple words while retains the ability of parallel computing.

5.7 Experiments

5.7.1 Datasets and Experiment Preparation

We conduct experiments on public datasets from SemEval workshops [PGP$^+$14], which consist of customer reviews about restaurants and laptops. Each sentence or aspect term can be labeled with multiple aspect categories and sentiments.

There are problems in the experiments of existing works. Some existing models such as IAN [MLZW17] and ATAE-LSTM [WHZZ16] remove “conflict” labels from four sentiment labels. Some removed sentences which have different sentiment labels for different aspects or targets in the sentence. However, this type of sentence is more popular in review data than in sentiment classification benchmark. After removing such complicated sentences, the resulting datasets have nothing special compared with standard sentence classification datasets, but no performance comparison with other neural sentiment classification model is provided, such as CNN [Kim14] and tree-LSTM [TSM15]. Moreover, such data preprocessing makes their results incomparable with the results of the workshop report. Therefore, we first reimplemented the compared methods and followed hyper-parameter settings described in these papers. Second, to access how the models perform on review sentences, we create small but difficult test datasets, denoted as SemEval2014-Mixed, which consists of the sentences having opposite or different sentiments on different aspects. For example in Table 5.1, the two identical sentences but with different sentiment labels are included in the dataset SemEval2014-Mixed.

Sentence	aspect category / aspect target	sentiment label
Average to good Thai food, but terrible delivery.	food	positive
Average to good Thai food, but terrible delivery.	delivery	negative

Table 5.1: Two example sentences in one test dataset **M** of restaurant review dataset of SemEval 2014.

Multiple aspect categories and aspect targets are labeled in the sentences and associated with one sentiment label. For ACSA task, we follow the paper of ATAE-LSTM [WHZZ16] and experiment on restaurant review data of SemEval 2014 Task 4. There are 5 aspects: *food*, *price*, *service*, *ambience*, and *misc* and 4 sentiment polarities: *positive*, *negative*, *neutral*, and *conflict*. By merging restaurant reviews of three years, we obtain a larger dataset called "Restaurant-Large". Incompatibilities of data are fixed during merging. The resulting dataset has 8 aspects: *restaurant*, *food*, *drinks*, *ambience*, *service*, *price*, *misc* and *location*; 3 sentiment polarities: *positive*, *negative*, and *neutral*. For ATSA task, we use restaurant reviews and laptop reviews from SemEval 2014 Task 4. On each dataset, we repeat each sentence n_a times, which is equal to the number of associated aspect categories (ACSA) or aspect terms (ATSA) [RGB16b, RGB16a]. The statistics of the unrolled datasets are shown in Table 5.2.

The sizes of data that have mixed sentiments are also shown in Table 5.2. It is designed to measure whether a model can detect multiple sentiment polarities in one sentence toward different entities; otherwise, on the sentences associated with only one sentiment, we can just use a classifier that is trained for overall sentiment classification.

	Positive		Negative		Neutral		Mixed	
	Train	Test	Train	Test	Train	Test	Train	Test
Restaurant-Large	2708	1505	1211	692	746	229	467	234

Table 5.2: The statistics of the unrolled dataset for ACSA task. The mixed subset is made of sentences having multiple aspect labels associated with multiple sentiments.

	Positive		Negative		Neutral		Conflict		Mixed	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Restaurant	2161	725	805	195	633	196	91	14	1038	244
Laptop	981	340	857	125	459	169	45	16	494	102

Table 5.3: The statistics of the unrolled dataset in SemEval 2014 for ATSA task.

We conduct experiments for ACSA task on restaurant reviews. Word embedding vectors are initialized with 300-dimension GloVe vectors which are pre-trained on unlabeled data of

840 billion tokens [PSM14]. Words out of the vocabulary of GloVe are randomly initialized with a uniform distribution $U(-0.25, 0.25)$. We use Adagrad [DHS11] with a batch size of 32 instances, default learning rate of $1e-2$, and maximal epochs of 30. We only fine tune early stopping with 5-fold cross validation on training datasets. All the models except SVM are implemented in PyTorch.

5.7.2 Compared Methods

To comprehensively evaluate the performance of GCAE, we compare our model against the following models.

NRC-Canada [KZCM14] is the top method in SemEval 2014 Task 4 for ACSA and ATSA task. SVM is trained with a number of features: n-grams, character n-grams, non-contiguous n-grams, POS tags, cluster n-grams, and lexicon features. The sentiment lexicons improve the performance significantly, but it requires large scale labeled data, such as 183K Yelp reviews, 124K Amazon laptop reviews, 56 million tweets, and three sentiment lexicons labeled manually.

CNN [Kim14] is widely used on text classification task. It cannot directly capture aspect-specific sentiment information on ACSA task, but it provides a very strong baseline for sentiment classification. We set the widths of filters to 3, 4, 5 with 100 features each.

GCN is gated convolutional neural network, in which GTRU does not have the aspect embedding as an additional input.

ATAE-LSTM [WHZZ16] is an attention-based LSTM for ACSA task. It appends the given aspect embedding with each word embedding as the input of LSTM, and has an attention layer above it.

IAN [MLZW17] stands for interactive attention network for ATSA task, which is also based on LSTM and attention mechanisms.

Models	Restaurant-Large		Restaurant 2014	
	T	M	T	M
SVM*	-	-	75.32	-
SVM + lexicons	-	-	**82.93**	-
ATAE-LSTM	83.91±0.49	66.32±2.28	78.29±0.68	45.62±0.90
CNN	84.28±0.15	50.43±0.38	79.47±0.32	44.94±0.01
GCN	84.48±0.06	50.08±0.31	**79.67±0.35**	44.49±1.52
GCAE	**85.92±0.27**	**70.75±1.19**	79.35±0.34	**50.55±1.83**

Table 5.4: The accuracy of all models on test datasets. (T) and the subsets made of test sentences with multiple sentiments (M) of restaurant reviews. Restaurant-Large dataset is created by merging all the restaurant reviews of SemEval workshops within three years. '*': the results with SVM are retrieved from NRC-Canada [KZCM14].

RAN [CSBY17] stands for recurrent attention network for ATSA task, which uses LSTM and multiple-attention mechanism. It also should be noted that the authors removed a lot of hard data records that have "conflict" labels, which are exactly the same as our difficult testing dataset M.

Although ATAE-LSTM, IAN, and RAN use SemEval datasets as well, their experiments are not exactly the same as the experiments required in the SemEval workshop. Therefore, we re-implement these three models and use the hyper-parameters described in their papers for fair comparison.

5.7.3 Results and Analysis

Following the SemEval workshop, we report the overall accuracy of all competing models over the test datasets of restaurant reviews as well as the harder subsets with mixed sentiments. Every experiment is repeated five times. The mean and the standard deviation are reported in Table 5.4, which are more sound than just one single accuracy score as in other existing work.

LSTM based model ATAE-LSTM has the worst performance of all the neural networks. Aspect-based sentiment analysis is to extract the sentiment information closely related to the given aspect. It is important to separate aspect information and sentiment information from sentences. The context vectors generated by LSTM have to convey the two kinds of information at the same time. Moreover, the attention scores generated by the similarity scoring function are for the entire context vector; while GTRU in our model can control the gates at each dimension of the context vector. By comparing the performance on the harder test dataset (M) against CNN, attention layer is able to distinguish different sentiments in one sentence but from different aspects.

Convolutional neural networks CNN and GCN, which are not designed for aspect based sentiment analysis, but they exceed the ATAE-LSTM by around 1%. It shows that CNN based networks are better on capturing sentiment information.

GCAE improves the performance by 1.1% to 2.5% compared with ATAE-LSTM. Our GCAE model incorporates GTRU to control the sentiment information flow according to the given aspect information. GTRU does not generate a single context vector like attention layer, but two vectors for aspect and sentiment features respectively. The element-wise gating mechanism works at fine granularity instead of exerting an alignment score to all the dimensions of the context vectors in attention layer.

The performance of SVM [KZCM14] depends on the availability of the features it can use. Without the large amount of sentiment lexicons, SVM perform worse than neural methods. With multiple sentiment lexicons, the performance is increased by 7.6%. This inspires our future work to leverage the sentiment lexicons in neural networks.

The test sets (M) consists of replicated sentences with different sentiments towards different aspects. The models which cannot utilize the given aspect information such as CNN and GCN perform poorly as expected, but GCAE still has higher accuracy than other neural network models, which proves the effectiveness of GTRU.

Kernel Widths	Accuracy
2	84.52
3	84.85
4	85.01
5	84.97
6	84.97
2, 3, 4	85.84
3, 4, 5	85.92
4, 5, 6	85.43
2, 3, 4, 5	85.76
2, 3, 4, 5, 6	85.98

Table 5.5: The accuracy of GCAE with convolutional kernels of different window sizes on ACSA task.

	Time (s)
ATAE-LSTM	25.07
CNN	**9.25**
GCN	9.43
GCAE	10.85

Table 5.6: The model training time in seconds on ACSA task.

5.7.4 Effects of Kernel Sizes

We investigate the effect of the window size of convolutional filters, as shown in Table 5.5. We fix the number of convolutional filter to 100. In general, using one type of filter is not as good as using more. Filter widths that are more than 3 do not bring significant improvement, and would have longer training time. Our model with three different filters works better.

5.7.5 Training Time

For ACSA task, we recorded the time of all models until convergence on a validation set on a desktop machine with a single Nvidia GeForce GTX 1080 Ti GPU and Intel Core i7-7600K Processor, as shown in Table 5.6 and Table 5.7. LSTM based models take more training time than convolutional models. On ACSA task, GCAE only spends less than half of time for training, compared with ATAE-LSTM. On ATSA task, because of multiple attention

	Time (s)
TD-LSTM	19.39
ATAE-LSTM	25.28
IAN	82.87
RAN	64.16
GCAE	**10.85**

Table 5.7: The model training time in seconds on ATSA task.

Gates	Restaurant-Large		Restaurant 2014	
	T	M	T	M
GTU	84.62	60.25	79.31	**51.93**
GLU	84.74	59.82	79.12	50.80
GTRU	**85.92**	**70.75**	**79.35**	50.55

Table 5.8: The accuracy of different gating units on restaurant reviews on ACSA task. T is the original test set, while M is the subset of test set which having mixed sentiments.

layers in IAN and RAN, they need even more time to finish the training. GCAE is just slightly slower than the vanilla CNN, and much faster than other neural models, because neither convolutional operation nor GTRU has data dependency, compared with LSTM and attention layer. Therefore, it is easier for hardware and library to parallel the computing process. Since the performance of SVM is retrieved from the original paper, we are not able to compare the training time of SVM.

5.7.6 Gating Mechanisms

In this section, we compare GLU $(\mathbf{X} * \mathbf{W} + b) \times \sigma(\mathbf{X} * \mathbf{W}_a + \mathbf{V}\boldsymbol{v}_a + b_a)$ [DFAG17], GTU $\tanh(\mathbf{X} * \mathbf{W} + b) \times \sigma(\mathbf{X} * \mathbf{W}_a + \mathbf{V}\boldsymbol{v}_a + b_a)$ [vdOKE+16], and GTRU used in GCAE. Table 5.8 shows that all of three gating units achieve relatively high accuracy on restaurant datasets. GTRU outperforms the other gates. It has a convolutional layer generating aspect features via ReLU activation function, which can control the magnitude of the sentiment signals according to the given aspect information. On the other hand, the sigmoid function

Models	Restaurant		Laptop	
	T	M	T	M
SVM*	77.13	-	63.61	-
SVM + lexicon	**80.16**	-	**70.49**	-
TDLSTM	73.44±1.17	56.48±2.46	62.23±0.92	46.11±1.89
ATAE-LSTM	73.74±3.01	50.98±2.27	64.38±4.52	40.39±1.30
IAN	76.34±0.27	55.16±1.43	68.49±0.57	44.51±0.48
RAN	76.51±0.94	53.55±1.62	67.00±3.28	42.41±2.83
GCAE	**77.28±0.32**	**56.73±0.56**	**69.14±0.32**	**47.06±2.45**

Table 5.9: The accuracy of ATSA subtask on SemEval 2014 Task 4. '*': the results with SVM are retrieved from NRC-Canada [KZCM14]

in GTU and GLU has the upper bound $+1$, which may not be able to distill sentiment features.

5.7.7 ATSA

We apply the extended version of GCAE on ATSA task. On this task, the aspect terms are marked in the sentences and usually consist of multiple words. We compare IAN [MLZW17], RAN [CSBY17], TDLSTM [TQFL16], ATAE-LSTM [WHZZ16], and our GCAE model in Table 5.9. The models other than GCAE is based on LSTM and attention mechanisms. IAN is better than TDLSTM and ATAE-LSTM, since it emphasizes the aspect terms with another attention layer. GCAE uses the outputs of the small CNN over aspect terms to guide the composition of the sentiment features through the ReLU gate. Because of the gating mechanisms and the modeling ability of aspect terms, GCAE outperforms other neural models and basic SVM. Again, large scale sentiment lexicons bring significant improvement to SVM.

5.8 Conclusions

In this chapter, we propose an efficient convolutional neural network with gating mechanisms for ACSA and ATSA tasks. The convolutional neurons can extract features faster than LSTM cells. The GTRU can effectively control the sentiment flow according to the given aspect information. We prove the performance improvement compared with other neural models by extensive experiments on SemEval datasets. How to leverage large-scale sentiment lexicons in neural networks would be our future work.

CHAPTER 6

Conclusion and Future Work

This dissertation develops machine learning models using probabilistic graphical models and neural networks for aspect based sentiment analysis on review data. New models are proposed for different subtasks in aspect based sentiment analysis to address challenges of fine-grained opinion mining on reviews. We highlight the contributions as follows

- We developed several topics models for aspect extraction and sentiment inference on hotel reviews, which can fully utilize the numerical features along the review text: overall sentiment polarity score and aspect sentiment polarity score. Numerical ratings as well as words are modeled as latent variables. The interdependencies among the probabilistic variables are accurately represented. Moreover, potential hierarchical structures of review data are also taken into account. The prediction performance of aspect ratings are increased.

- Aspect category classification and aspect term extraction are simultaneously handled within a single neural network. The traditional models often rely on laborious manual feature engineering. Neural networks can automatically learn efficient feature representations for semantic and syntactic information within sentences. We apply convolutional neural networks and long short term memory to address the two subtasks in a way of multi-task learning. The latent representations are shared between two tasks, which greatly reduces the noise of each task and increases both prediction accuracy.

- Aspect sentiment polarity classification is an essential task at sentence level and at aspect level. Based on convolutional networks, we proposed a gated ReLU-Tanh unit which can selectively predict sentiment polarity according to the given aspect. It has very impressive training speed and simple structure compared to existing models that use long short-term memory and attention mechanisms.

In summary, this dissertation demonstrates the powerfulness of graphical models and neural networks for various tasks in aspect based sentiment analysis. Review data as a kind of social media data plays an important role in online activities in modern world. There are many other questions remain unsolved. Neural networks gains much popularity in recent research because of its great representation and modeling ability, but the question that whether neural networks can produce a coherent and meaningful summary of given review data needs further research effort.

BIBLIOGRAPHY

[AAB$^+$16] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Gregory S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian J Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Gordon Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul A Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda B Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow - Large-Scale Machine Learning on Heterogeneous Distributed Systems. *CoRR*, abs/1603.04467, 2016.

[AEHKL$^+$16] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. YouTube-8M: A Large-Scale Video Classification Benchmark. September 2016.

[AMCG15] Abdulaziz Alghunaim, Mitra Mohtarami, Scott Cyphers, and Jim Glass. A Vector Space Approach for Aspect Based Sentiment Analysis. In *VS@HLT-NAACL*, pages 116–122, 2015.

[BBB10] J Bergstra, O Breuleux, and F Bastien. Theano: A CPU and GPU math compiler in Python. *Proceedings of the 9th Python in Science Conference*, 2010.

[BCB14] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR*, abs/1409.0473, 2014.

[BDVJ03] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3(Feb):1137–1155, 2003.

[Bis07] Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.

[BKH16] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer Normalization. *CoRR*, abs/1607.06450, July 2016.

[BLPL06] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy Layer-Wise Training of Deep Networks. In *NIPS*, pages 153–160, 2006.

[BLT17] Konstantin Bauman, Bing Liu, and Alexander Tuzhilin. Aspect Based Recommendations - Recommending Items with the Most Valuable Aspects Based on User Reviews. In *KDD*, pages 717–725. ACM Press, 2017.

[BMXS16] James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher. Quasi-Recurrent Neural Networks. In *ICLR*, pages abs–1611.01576, November 2016.

[BNJ03] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022, 2003.

[BSF94] Yoshua Bengio, P Simard, and P Frasconi. Learning long-term dependencies with gradient descent is difficult . *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[CGCB14] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. In *NIPS*, pages abs–1412.3555, 2014.

[Cho16] François Chollet. Xception - Deep Learning with Depthwise Separable Convolutions. In *CVPR*, pages abs–1610.02357, 2016.

[CK11] R Collobert and K Kavukcuoglu. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011.

[CLSZ13] Zheng Chen, Chengtao Li, Jian-Tao Sun, and Jianwen Zhang. Sentiment Topic Model with Decomposed Prior. In *SDM*, pages 767–775, Philadelphia, PA, 2013. Society for Industrial and Applied Mathematics.

[CN15] Jason P. C. Chiu and Eric Nichols. Named Entity Recognition with Bidirectional LSTM-CNNs. *TACL*, 4:357–370, 2015.

[CQZ$^+$15] Xinchi Chen, Xipeng Qiu, Chenxi Zhu, Shiyu Wu, and Xuanjing Huang. Sentence Modeling with Gated Recursive Neural Network. In *EMNLP*, pages 793–798, 2015.

[CSBL16] Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann LeCun. Very Deep Convolutional Networks for Text Classification. In *EACL*, pages 1107–1116, June 2016.

[CSBY17] Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. Recurrent Attention Network on Memory for Aspect Sentiment Analysis. In *EMNLP*, pages 463–472, 2017.

[CST$^+$16] Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. Neural Sentiment Classification with User and Product Attention. In *EMNLP*, pages 1650–1659, 2016.

[CUH16] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). In *ICLR*, pages abs–1511.07289, 2016.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[CvMG$^+$14] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *EMNLP*, pages 1724–1734, 2014.

[CW08] Ronan Collobert and Jason Weston. A unified architecture for natural language processing - deep neural networks with multitask learning. In *ICML*, pages 160–167, New York, New York, USA, 2008. ACM Press.

[CWB$^+$11] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P Kuksa. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

[CZLZ16] Wenliang Chen, Zhenjie Zhang, Zhenghua Li, and Min Zhang. Distributed Representations for Building Profiles of Users and Items from Text Reviews. In *COLING*, pages 1724–1734, 2016.

[DC04] Sanjiv Ranjan Das and Mike Y Chen. Yahoo! for Amazon: Sentiment Parsing from Small Talk on the Web. *SSRN Electronic Journal*, pages 1375–1388, 2004.

[DDS^{+}09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet - A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[DFAG17] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language Modeling with Gated Convolutional Networks. In *ICML*, pages 933–941, 2017.

[DHS11] John C Duchi, Elad Hazan, and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, pages 2121–2159, 2011.

[DHW^{+}17] Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. Learning to Generate Product Reviews from Attributes. In *EACL*, pages 623–632, 2017.

[dLFL16] Bart de Langhe, Philip M Fernbach, and Donald R Lichtenstein. Navigating by the Stars: Investigating the Actual and Perceived Validity of Online User Ratings. *Journal of Consumer Research*, 42(6):817–833, April 2016.

[DLP03] Kushal Dave, Steve Lawrence, and David M Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *WWW*, pages 519–528, New York, New York, USA, May 2003. ACM.

[DQW^{+}14] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *KDD*, pages 193–202, New York, New York, USA, 2014. ACM Press.

[dSG14] Cícero Nogueira dos Santos and Maira Gatti. Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. In *COLING*, pages 69–78, 2014.

[DWT^{+}14] Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. Adaptive Recursive Neural Network for Target-dependent Twitter Sentiment Classification. In *ACL*, pages 49–54, 2014.

[Elm90] Jeffrey L Elman. Finding Structure in Time. *Cognitive Science*, 14(2):179–211, 1990.

[Fir57] John Rupert Firth. A synopsis of linguistic theory 1930-1955. In *Studies in Linguistic Analysis*. 1957.

[GAG$^+$17] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional Sequence to Sequence Learning. In *ICML*, pages 1243–1252, 2017.

[GBC16] Ian Goodfellow, Yoshua Bengio, and A Courville. *Deep learning*. MIT Press, 2016.

[GCZ$^+$16] Ziyu Guan, Long Chen, Wei Zhao, Yi Zheng, Shulong Tan, and Deng Cai. Weakly-Supervised Deep Learning for Customer Review Sentiment Classification. In *IJCAI*, pages 3719–3725, 2016.

[GG16] Yarin Gal and Zoubin Ghahramani. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. In *NIPS*, pages 1019–1027, 2016.

[Gra12] Alex Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*. PhD thesis, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[GS04] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235, April 2004.

[GS05] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602–610, July 2005.

[GWD14] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural Turing Machines. *CoRR*, abs/1410.5401, October 2014.

[GWFM$^+$13] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout Networks. In *ICML*, pages 1319–1327, February 2013.

[GWK$^+$15] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, and Gang Wang. Recent Advances in Convolutional Neural Networks. *CoRR*, abs/1512.07108, December 2015.

[GX13] Yuhong Guo and Wei Xue. Probabilistic Multi-Label Classification with Sparse Feature Learning. In *IJCAI*, pages 1373–1379, 2013.

[HCKC15] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. TriRank - Review-aware Explainable Recommendation by Modeling Aspects. In *CIKM*, pages 1661–1670, New York, New York, USA, 2015. ACM Press.

[Hin84] G E Hinton. Distributed representations. In *Parallel Distributed Processing*. 1984.

[HL04] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *KDD*, pages 168–177, New York, New York, USA, 2004. ACM Press.

[HMJS17] Shiou Tian Hsu, Changsung Moon, Paul Jones, and Nagiza F Samatova. A Hybrid CNN-RNN Alignment Model for Phrase-Aware Sentence Classification. In *EACL*, pages 443–449, 2017.

[Hof99] Thomas Hofmann. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–57, New York, New York, USA, 1999. ACM Press.

[Hol16] Elizabeth Holmes. When Shopping Online, Can You Trust the Reviews?, 2016.

[HOT06] G E Hinton, S Osindero, and Y W Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[HS] G E Hinton and T J Sejnowski. Learning and relearning in Boltzmann machines. In *Parallel Distributed Processing*. cs.toronto.edu.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural computation*, 9(8):1735–1780, 1997.

[HSMN12] Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. Improving Word Representations via Global Context and Multiple Word Prototypes. In *ACL*, pages 873–882, 2012.

[HW59] D H Hubel and T N Wiesel. Receptive fields of single neurones in the cat&apos;s striate cortex. *The Journal of Physiology*, 148(3):574–591, October 1959.

[HZRS15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *ICCV*, pages 1026–1034. IEEE, 2015.

[HZRS16a] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, pages 770–778, 2016.

[HZRS16b] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity Mappings in Deep Residual Networks. In *ECCV*, pages 630–645, March 2016.

[IC14] Ozan Irsoy and Claire Cardie. Opinion Mining with Deep Recurrent Neural Networks. In *EMNLP*, pages 720–728, 2014.

[IMBGDI15] Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. Deep Unordered Composition Rivals Syntactic Methods for Text Classification. In *ACL*, pages 1681–1691, 2015.

[IS15] Sergey Ioffe and Christian Szegedy. Batch Normalization - Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *ICML*, pages 448–456, 2015.

[JO11] Yohan Jo and Alice H Oh. Aspect and sentiment unification model for online review analysis. In *WSDM*, pages 815–824, New York, New York, USA, 2011. ACM Press.

[JSD$^+$14] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678, 2014.

[JYP$^+$17] Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, Rick Boyle, Pierre-luc Cantin, Clifford Chao, Chris Clark, Jeremy Coriell, Mike Daley, Matt Dau, Jeffrey Dean, Ben Gelb, Tara Vazir Ghaemmaghami, Rajendra Gottipati, William Gulland, Robert Hagmann, C Richard Ho, Doug Hogberg, John Hu, Robert Hundt, Dan Hurt, Julian Ibarz, Aaron Jaffey, Alek Jaworski, Alexander Kaplan, Harshit Khaitan, Andy Koch, Naveen Kumar, Steve Lacy, James Laudon, James Law, Diemthu Le, Chris Leary, Zhuyuan Liu, Kyle Lucke, Alan Lundin, Gordon MacKean, Adriana Maggiore, Maire Mahony, Kieran Miller, Rahul Nagarajan, Ravi Narayanaswami, Ray Ni, Kathy Nix, Thomas Norrie, Mark Omernick, Narayana Penukonda, Andy Phelps, Jonathan Ross, Matt Ross, Amir Salek, Emad Samadiani, Chris Severn, Gregory Sizikov, Matthew Snelham, Jed Souter, Dan Steinberg, Andy Swing, Mercedes Tan, Gregory Thorson, Bo Tian, Horia Toma, Erick Tuttle, Vijay Vasudevan, Richard Walter, Walter Wang, Eric Wilcox, and Doe Hyun Yoon. In-Datacenter Performance Analysis of a Tensor Processing Unit. In *International Symposium on Computer Architecture*, pages 1–12, April 2017.

[JZ15] Rie Johnson and Tong Zhang. Effective Use of Word Order for Text Categorization with Convolutional Neural Networks. In *NAACL-HLT*, pages abs–1412.1058, 2015.

[JZ17] Rie Johnson and Tong Zhang. Deep Pyramid Convolutional Neural Networks for Text Categorization. In *ACL*, pages 562–570, Stroudsburg, PA, USA, 2017. Association for Computational Linguistics.

[KB15] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015.

[KEB16] Talaat Khalil and Samhaa R El-Beltagy. NileTMRG at SemEval-2016 Task 5 - Deep Convolutional Neural Networks for Aspect Category and Sentiment Extraction. *SemEval@NAACL-HLT*, pages 271–276, 2016.

[KES$^+$16] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aäron van den Oord, Alex Graves, and Koray Kavukcuoglu. Neural Machine Translation in Linear Time . *CoRR*, abs/1610.10099, October 2016.

[KF09] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models*. MIT Press, 2009.

[KGB14] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In *ACL*, pages 655–665, 2014.

[Kim14] Yoon Kim. Convolutional Neural Networks for Sentence Classification. In *EMNLP*, pages 1746–1751, 2014.

[KLMR16] Ben Krause, Liang Lu, Iain Murray, and Steve Renals. Multiplicative LSTM for sequence modelling. *CoRR*, abs/1609.07959, September 2016.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, pages 1106–1114, 2012.

[KUMH17] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-Normalizing Neural Networks. *CoRR*, abs/1706.02515, 2017.

[KZCM14] Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif M. Mohammad. NRC-Canada-2014: Detecting aspects and sentiment in customer reviews. In *SemEval@COLING*, pages 437–442, Stroudsburg, PA, USA, 2014. Association for Computational Linguistics.

[KZS$^+$15] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-Thought Vectors. In *NIPS*, pages 3294–3302, 2015.

[LBBH98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *IEEE*, 86(11):2278–2324, 1998.

[LBD$^+$89] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, R E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten Digit Recognition with a Back-Propagation Network. In *NIPS*, pages 396–404, 1989.

[LBOM12] Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient BackProp. *Neural Networks - Tricks of the Trade*, 7700(2):9–48, 2012.

[LBS$^+$16] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural Architectures for Named Entity Recognition. In *VS@HLT-NAACL*, pages 260–270, March 2016.

[LCD17] Hoa T Le, Christophe Cerisara, and Alexandre Denis. Do Convolutional Networks need to be Deep for Text Classification ? *CoRR*, abs/1707.04108, July 2017.

[LG14] Omer Levy and Yoav Goldberg. Neural Word Embedding as Implicit Matrix Factorization. In *NIPS*, pages 2177–2185, 2014.

[LGD15] Omer Levy, Yoav Goldberg, and Ido Dagan. Improving Distributional Similarity with Lessons Learned from Word Embeddings. In *ACL*, pages 211–225, May 2015.

[LGM17] Cheng Li, Xiaoxiao Guo, and Qiaozhu Mei. Deep Memory Networks for Attitude Identification. In *WSDM*, pages 671–680. ACM Press, 2017.

[LH09] Chenghua Lin and Yulan He. Joint sentiment/topic model for sentiment analysis. In *CIKM*, pages 375–384, New York, New York, USA, 2009. ACM Press.

[LHB15] Yann LeCun, Geoffrey Hinton, and Yoshua Bengio. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[LHLN15] Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. Joint Entity Recognition and Disambiguation. In *EMNLP*, pages 879–888, Stroudsburg, PA, USA, 2015. Association for Computational Linguistics.

[LJM15] Pengfei Liu, Shafiq R Joty, and Helen M Meng. Fine-grained Opinion Mining with Recurrent Neural Networks and Word Embeddings. In *EMNLP*, pages 1433–1443, 2015.

[LLJH15] Jiwei Li, Thang Luong, Dan Jurafsky, and Eduard H Hovy. When Are Tree Structures Necessary for Deep Learning of Representations? In *EMNLP*, pages abs–1503.00185, 2015.

[LLS03] Hugo Liu, Henry Lieberman, and Ted Selker. A model of textual affect sensing using real-world knowledge. In *IUI*, pages 125–132, New York, New York, USA, January 2003. ACM.

[LM14] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *ICML*, pages 1188–1196, 2014.

[LMP01] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289, 2001.

[LPM15] Thang Luong, Hieu Pham, and Christopher D Manning. Effective Approaches to Attention-based Neural Machine Translation. In *EMNLP*, pages 1412–1421, 2015.

[LQH16] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Deep Multi-Task Learning with Shared Memory for Text Classification. In *EMNLP*, pages 118–127, 2016.

[LQH17] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Dynamic Compositional Neural Networks over Tree Structure. In *IJCAI*, pages 4054–4069, 2017.

[LSM14] Himabindu Lakkaraju, Richard Socher, and Chris Manning. Aspect specific sentiment analysis using hierarchical deep learning. In *NIPS Workshop on Deep Learning and Representation Learning*, 2014.

[LXLZ15] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent Convolutional Neural Networks for Text Classification. *AAAI*, pages 2267–2273, 2015.

[LZ12] Bing Liu and Lei Zhang. A Survey of Opinion Mining and Sentiment Analysis. *Mining Text Data*, (Chapter 13):415–463, 2012.

[LZC$^+$14] Wenjuan Luo, Fuzhen Zhuang, Xiaohu Cheng, Qing He, and Zhongzhi Shi. Ratable Aspects over Sentiments - Predicting Ratings for Unrated Reviews. In *ICDM*, pages 380–389. IEEE, 2014.

[LZL$^+$16] Bofang Li, Zhe Zhao, Tao Liu, Puwei Wang, and Xiaoyong Du. Weighted Neural Bag-of-n-grams Model - New Baselines for Text Classification. In *COLING*, pages 1591–1600, 2016.

[LZS09] Yue Lu, ChengXiang Zhai, and Neel Sundaresan. Rated aspect summarization of short comments. In *WWW*, pages 131–140, New York, New York, USA, 2009. ACM Press.

[Mac16] Jakub Machacek. BUTknot at SemEval-2016 Task 5 - Supervised Machine Learning with Term Substitution Approach in Aspect Category Detection. *SemEval@NAACL-HLT*, pages 301–305, 2016.

[MCCD13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In *ICLR*, pages abs–1301.3781, 2013.

[ME11] Samaneh Moghaddam and Martin Ester. ILDA - interdependent LDA model for learning latent aspects and their ratings from online product reviews. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 665–674, New York, New York, USA, 2011. ACM Press.

[ME12] Samaneh Moghaddam and Martin Ester. On the design of IDA models for aspect-based opinion mining. In *CIKM*, pages 803–812, New York, New York, USA, 2012. ACM Press.

[MH16] Xuezhe Ma and Eduard Hovy. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *ACL*, pages 1064–1074, March 2016.

[MHN13] Andrew L Mass, Awni Y Hannun, and Andrew Y Ng. Rectifier Nonlinearities Improve Neural Network Acoustic Models . In *ICML*, 2013.

[MHZX15] Mingbo Ma, Liang Huang, Bowen Zhou, and Bing Xiang. Dependency-based Convolutional Neural Networks for Sentence Embedding. In *ACL*, pages 174–179, 2015.

[MLW$^+$07] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. Topic sentiment mixture - modeling facets and opinions in weblogs. In *WWW*, pages 171–180, New York, New York, USA, 2007. ACM Press.

[MLZW17] Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. Interactive Attention Networks for Aspect-Level Sentiment Classification. In *IJCAI*, pages 4068–4074. International Joint Conferences on Artificial Intelligence Organization, 2017.

[MSC$^+$13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*, pages 3111–3119, 2013.

[MYTF02] Satoshi Morinaga, Kenji Yamanishi, Kenji Tateishi, and Toshikazu Fukushima. Mining product reputations on the Web. In *KDD*, pages 341–349, New York, New York, USA, July 2002. ACM.

[NBGS08] John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. Scalable Parallel Programming with CUDA. *ACM Queue*, 6(2):40, 2008.

[Nes83] Yurii Nesterov. A method of solving a convex programming problem with convergence rate O (1/k2). In *Soviet Mathematics Doklady*, pages 372–376, 1983.

[NH10] V Nair and G E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010.

[NS15] Thien Hai Nguyen and Kiyoaki Shirai. PhraseRNN: Phrase Recursive Neural Network for Aspect-based Sentiment Analysis. In *EMNLP*, pages 2509–2514, 2015.

[NY03] Tetsuya Nasukawa and Jeonghee Yi. Sentiment analysis: capturing favorability using natural language processing. In *K-CAP*, pages 70–77, New York, New York, USA, October 2003. ACM.

[O'C10] Peter O'Connor. Managing a Hotel's Image on TripAdvisor. *Journal of Hospitality Marketing & Management*, 19(7):754–772, September 2010.

[PGP$^+$14] Maria Pontiki, Dimitrios Galanis, John Pavlopoulos, Haris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. Semeval-2014 task 4: Aspect based sentiment analysis. In *SemEval@COLING*, pages 27–35, Stroudsburg, PA, USA, 2014. Association for Computational Linguistics.

[PGP$^+$15] Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. Semeval-2015 task 12: Aspect based sentiment analysis. In *SemEval 2015*, pages 486–495, Stroudsburg, PA, USA, 2015. Association for Computational Linguistics.

[PGP$^+$16] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphee De Clercq, Veronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Núria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. SemEval-2016 Task 5: Aspect Based Sentiment Analysis. In *SemEval@NAACL-HLT*, pages 19–30, Stroudsburg, PA, USA, 2016. Association for Computational Linguistics.

[PL05] Bo Pang and Lillian Lee. Seeing stars. In *ACL*, pages 115–124, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[PL08] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieva*, 2:1–135, 2008.

[PLV02] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *ACL*, pages 79–86, May 2002.

[PMB13] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *ICML*, pages 1310–1318, 2013.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global Vectors for Word Representation. In *EMNLP*, pages 1532–1543, 2014.

[QHLZ17] Qiao Qian, Minlie Huang, Jinhao Lei, and Xiaoyan Zhu. Linguistically Regularized LSTM for Sentiment Classification. In *ACL*, pages 1679–1689. Association for Computational Linguistics, 2017.

[Qia99] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1):145–151, 1999.

[QLBC11] Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. Opinion Word Expansion and Target Extraction through Double Propagation. *Computational Linguistics*, 37(1):9–27, 2011.

[QTH$^+$15] Qiao Qian, Bo Tian, Minlie Huang, Yang Liu, Xuan Zhu, and Xiaoyan Zhu. Learning Tag Embeddings and Tag-specific Composition Functions in Recursive Neural Network. In *ACL*, pages 1365–1374, 2015.

[RCW15] Alexander M Rush, Sumit Chopra, and Jason Weston. A Neural Attention Model for Abstractive Sentence Summarization. In *EMNLP*, pages 379–389, 2015.

[RGB16a] Sebastian Ruder, Parsa Ghaffari, and John G Breslin. A Hierarchical Model of Reviews for Aspect-based Sentiment Analysis. In *EMNLP*, pages 999–1005, September 2016.

[RGB16b] Sebastian Ruder, Parsa Ghaffari, and John G Breslin. INSIGHT-1 at SemEval-2016 Task 5 - Deep Learning for Multilingual Aspect-based Sentiment Analysis. In *SemEval@NAACL-HLT*, pages 330–336, 2016.

[RHW] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. In *Parallel Distributed Processing*.

[RHW86] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[RJS17] Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. Learning to Generate Reviews and Discovering Sentiment. *CoRR*, abs/1704.01444, April 2017.

[RMN09] Rajat Raina, Anand Madhavan, and Andrew Y Ng. Large-scale deep unsupervised learning using graphics processors. In *ICML*, pages 1–8, New York, New York, USA, 2009. ACM Press.

[RR09] Lev-Arie Ratinov and Dan Roth. Design Challenges and Misconceptions in Named Entity Recognition. In *CoNLL*, pages 147–155, 2009.

[Rud16] Sebastian Ruder. An overview of gradient descent optimization algorithms. *CoRR*, arXiv:1609.04747v2, 2016.

[Rud17] Sebastian Ruder. An Overview of Multi-Task Learning in Deep Neural Networks. *CoRR*, abs/1706.05098, 2017.

[SB07] Benjamin Snyder and Regina Barzilay. Multiple Aspect Ranking Using the Good Grief Algorithm. In *NAACL-HLT*, pages 300–307, 2007.

[SHK$^+$14] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout - a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, pages 1929–1958, 2014.

[SHMN12] Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. Semantic Compositionality through Recursive Matrix-Vector Spaces. In *EMNLP*, pages 1201–1211, 2012.

[SLJ$^+$15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9. IEEE, 2015.

[SMH11] Ilya Sutskever, James Martens, and Geoffrey E Hinton. Generating Text with Recurrent Neural Networks. In *ICML*, pages 1017–1024, 2011.

[SP97] M Schuster and K K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[SPH$^+$11] Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions. In *EMNLP*, pages 151–161, 2011.

[SPWC13] Richard Socher, Alex Perelygin, Jean Y Wu, and Jason Chuang. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, pages 1631–1642, 2013.

[SSWF15] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-To-End Memory Networks. In *NIPS*, pages 2440–2448, 2015.

[Ste16] Laura Stevens. Survey Shows Rapid Growth in Online Shopping, 2016.

[SVI$^+$16] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *CVPR*, pages 2818–2826. IEEE, 2016.

[SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to Sequence Learning with Neural Networks. In *NIPS*, pages 3104–3112, 2014.

[SZ15] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, pages abs–1409.1556, 2015.

[TM08a] Ivan Titov and Ryan T McDonald. A Joint Model of Text and Aspect Ratings for Sentiment Summarization. In *ACL*, pages 308–316, 2008.

[TM08b] Ivan Titov and Ryan T McDonald. Modeling online reviews with multi-grain topic models. In *WWW*, pages 111–120, 2008.

[TQFL16] Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. Effective LSTMs for Target-Dependent Sentiment Classification. In *COLING*, pages 3298–3307, 2016.

[TQL15a] Duyu Tang, Bing Qin, and Ting Liu. Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. In *EMNLP*, pages 1422–1432, 2015.

[TQL15b] Duyu Tang, Bing Qin, and Ting Liu. Learning Semantic Representations of Users and Products for Document Level Sentiment Classification. In *ACL*, pages 1014–1023. ACL, 2015.

[TQL16] Duyu Tang, Bing Qin, and Ting Liu. Aspect Level Sentiment Classification with Deep Memory Network. In *EMNLP*, pages 214–224, May 2016.

[TQLY15] Duyu Tang, Bing Qin, Ting Liu, and Yuekui Yang. User Modeling with Neural Network for Review Rating Prediction. In *IJCAI*, pages 1340–1346, 2015.

[TS15] Zhiqiang Toh and Jian Su. NLANGP: Supervised Machine Learning System for Aspect Category Classification and Opinion Target Extraction. In *SemEval@NAACL-HLT*, pages 496–501, 2015.

[TS16] Zhiqiang Toh and Jian Su. NLANGP at SemEval-2016 Task 5: Improving Aspect Based Sentiment Analysis using Neural Network Features. In *SemEval@NAACL-HLT*, pages 282–288, 2016.

[TSM15] Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In *ACL*, pages 1556–1566, 2015.

[Tur02] Peter D Turney. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In *ACL*, pages 417–424, 2002.

[TW14] Zhiqiang Toh and Wenting Wang. DLIREC: Aspect Term Extraction and Term Polarity Classification System. In *SemEval@COLING*, pages 235–240, Stroudsburg, PA, USA, 2014. Association for Computational Linguistics.

[TYC$^+$16] Jian Tang, Yifan Yang, Samuel Carton, Ming Zhang, and Qiaozhu Mei. Context-aware Natural Language Generation with Recurrent Neural Networks. *CoRR*, abs/1611.09900, 2016.

[vdOKE$^+$16] Aäron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Koray Kavukcuoglu, Oriol Vinyals, and Alex Graves. Conditional Image Generation with PixelCNN Decoders. In *NIPS*, pages 4790–4798, 2016.

[WCB14] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory Networks. In *ICLR*, pages CoRR abs–1410.3916, October 2014.

[WHZZ16] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based LSTM for Aspect-level Sentiment Classification. In *EMNLP*, pages 606–615, 2016.

[WJL16] Xingyou Wang, Weijie Jiang, and Zhiyong Luo. Combination of Convolutional and Recurrent Neural Network for Sentiment Analysis of Short Texts. In *COLING*, pages 2428–2437, 2016.

[WL16] Lu Wang and Wang Ling. Neural Network-Based Abstract Generation for Opinions and Arguments. In *NAACL-HLT*, pages 47–57, 2016.

[WLZ10] Hongning Wang, Yue Lu, and ChengXiang Zhai. Latent aspect rating analysis on review text data - a rating regression approach. In *KDD*, pages 783–792, New York, New York, USA, 2010. ACM Press.

[WM12] Sida I Wang and Christopher D Manning. Baselines and Bigrams - Simple, Good Sentiment and Topic Classification. In *ACL*, pages 90–94, 2012.

[WPDX16] Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. Recursive Neural Conditional Random Fields for Aspect-based Sentiment Analysis. In *EMNLP*, pages 616–626, 2016.

[WPDX17] Weya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. Coupled Multi-Layer Attentions for Co-Extraction of Aspect and Opinion Terms. In *AAAI*, pages 3316–3322, 2017.

[WYLZ16] Jin Wang, Liang-Chih Yu, K Robert Lai, and Xue-jie Zhang. Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model. In *ACL*, pages 225–230, 2016.

[XBK$^+$15] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *ICML*, pages 2048–2057, February 2015.

[XCQH16] Jiacheng Xu, Danlu Chen, Xipeng Qiu, and Xuanjing Huang. Cached Long Short-Term Memory Neural Networks for Document-Level Sentiment Classification. In *EMNLP*, pages 1660–1669, 2016.

[XLR15] Wei Xue, Tao Li, and Naphtali Rishe. Aspect and Ratings Inference with Aspect Ratings - Supervised Generative Models for Mining Hotel Reviews. In *WISE*, pages 17–31, Cham, 2015. Springer International Publishing.

[YC13] Bishan Yang and Claire Cardie. Joint Inference for Fine-grained Opinion Extraction. In *ACL*, pages 1640–1649, 2013.

[YSZ17a] Ali Yadollahi, Ameneh Gholipour Shahraki, and Osmar R Zaiane. Current State of Text Sentiment Analysis from Opinion to Emotion Mining. *ACM Computing Surveys*, 50(2):1–33, June 2017.

[YSZ17b] Yichun Yin, Yangqiu Song, and Ming Zhang. Document-Level Multi-Aspect Sentiment Classification as Machine Comprehension. In *EMNLP*, pages 2044–2054, 2017.

[YTW$^+$17] Min Yang, Wenting Tu, Jingxuan Wang, Fei Xu, and Xiaojun Chen. Attention Based LSTM for Target Dependent Sentiment Classification. In *AAAI*, pages 5013–5014, 2017.

[YWD$^+$16] Yichun Yin, Furu Wei, Li Dong, Kaimeng Xu, Ming Zhang, and Ming Zhou. Unsupervised Word and Dependency Path Embeddings for Aspect Term Extraction. In *IJCAI*, pages 2979–2985, 2016.

[YWLZ17] Liang-Chih Yu, Jin Wang, K Robert Lai, and Xue-jie Zhang. Refining Word Embeddings for Sentiment Analysis. In *EMNLP*, pages 534–539, 2017.

[YYD$^+$16] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J Smola, and Eduard H Hovy. Hierarchical Attention Networks for Document Classification. In *NAACL-HLT*, pages 1480–1489, 2016.

[Zei12] Matthew D Zeiler. ADADELTA: An Adaptive Learning Rate Method. *CoRR*, abs/1212.5701, December 2012.

[ZJYL10] Wayne Xin Zhao, Jing Jiang, Hongfei Yan, and Xiaoming Li. Jointly Modeling Aspects and Opinions with a MaxEnt-LDA Hybrid. In *EMNLP*, pages 56–65, 2010.

[ZLR16] Rui Zhang, Honglak Lee, and Dragomir Radev. Dependency Sensitive Convolutional Neural Networks for Modeling Sentences and Documents. In *VS@HLT-NAACL*, pages 1512–1521, November 2016.

[ZLZ$^+$14] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 83–92, New York, New York, USA, 2014. ACM Press.

[ZQZ$^+$16] Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling. In *COLING*, pages 3485–3495, 2016.

[ZRW16] Ye Zhang, Stephen Roller, and Byron C Wallace. MGNC-CNN - A Simple Approach to Exploiting Multiple Word Embeddings for Sentence Classification. In *NAACL-HLT*, pages 1522–1527, 2016.

[ZW15] Ye Zhang and Byron Wallace. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. *CoRR*, abs/1510.03820, 2015.

[ZWL18] Lei Zhang, Shuai Wang, and Bing Liu. Deep Learning for Sentiment Analysis : A Survey. *CoRR*, abs/1801.07883, January 2018.

[ZZL15] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level Convolutional Networks for Text Classification. In *NIPS*, pages 649–657, 2015.

[ZZV16] Meishan Zhang, Yue Zhang, and Duy-Tin Vo. Gated Neural Networks for Targeted Sentiment Analysis. In *AAAI*, pages 3087–3093, 2016.

VITA

WEI XUE

2008	B.S., Computer Science Zhejiang University Hangzhou, P.R. China
2011	M.S., Computer Science Zhejiang University Hangzhou, P.R. China
2012–2018	Ph.D., Computer Science Florida International University Miami, FL, USA

PUBLICATIONS AND PRESENTATIONS

- Wei Xue, Tao Li, and Naphtali Rishe. "Aspect identification and ratings inference for hotel reviews." *World Wide Web*, 2(1), pp. 23-37, 2017.
- Wei Xue, Tao Li, and Naphtali Rishe. "Aspect and ratings inference with aspect ratings: Supervised generative models for mining hotel reviews." In *International Conference on Web Information Systems Engineering*, pp. 17-31, 2015.
- Wei Xue, Wubai Zhou, Tao Li and Qing Wang. "MTNA: A Neural Multi-task Model for Aspect Category ClassiïňĄcation and Aspect Term Extraction On Restaurant Reviews." In *The 8th International Joint Conference on Natural Language Processing (IJCNLP)*, pp. 151-156, 2017.
- Wei Xue and Tao Li. "Aspect Based Sentiment Analysis with Gated Convolutional Networks." In *The 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.
- Wubai Zhou, Wei Xue, Tao Li, Chunqiu Zeng and Wang Qing."STAR: A System for Ticket Analysis and Resolution." In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017.
- Qifeng Zhou, Tao Li, Wei Xue, Chunqiu Zeng, Bin Xia, Ruyuan Han, Linkai Luo. "An Advanced Inventory Data Mining System for Business Intelligence." In *Big Data Computing Service and Applications (BigDataService)*, 2017, pp 210-217.
- Tao Li, Chunqiu Zeng, Wubai Zhou, Wei Xue, Yue Huang, Zheng Liu, Qifeng Zhou, Bin Xia, Qing Wang, Wentao Wang, Xiaolong Zhu. "FIU-Miner (a fast, integrated, and user-friendly system for data mining) and its applications" In *Knowledge and Information Systems*, August 2017, Volume 52, Issue 2, pp 411-443.

- Tao Li, Ning Xie, Chunqiu Zeng, Wubai Zhou, Li Zheng, Yexi Jiang, Yimin Yang, Hsin-yu Ha, Wei Xue, Yue Huang, Shu-ching Chen, Jainendra Navlakha, And S. S. Iyengar. “Data-Driven Techniques in Disaster Information Management.” *ACM Computing Surveys (CSUR)*, Volume 50, no. 1 (2017): 1.
- Li Zheng, Chunqiu Zeng, Lei Li, Yexi Jiang, Wei Xue, Jingxuan Li, Chao Shen et al. “Applying data mining techniques to address critical process optimization needs in advanced manufacturing.” In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1739-1748. ACM, 2014
- Guo, Yuhong, and Wei Xue. “Probabilistic Multi-Label Classification with Sparse Feature Learning.” In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1373-1379. 2013.