A Joint Model of Term Extraction and Polarity Classification for Aspect-based Sentiment Analysis

NGUYEN Ngoc Gia Hy

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology

September, 2018

Master's Thesis

A Joint Model of Term Extraction and Polarity Classification for Aspect-based Sentiment Analysis

1610434 NGUYEN Ngoc Gia Hy

Supervisor	Kiyoaki Shirai
Main Examiner	Kiyoaki Shirai
Examiners	Minh Le Nguyen
	Shinobu Hasegawa
	Kokoro Ikeda

Graduate School of Advanced Science and Technology

Japan Advanced Institute of Science and Technology

[Information Science]

August, 2018

Abstract

With explosion of the Internet in this digital age, more and more people easily access the Internet and share their opinions on social media (e.g., reviews, forum discussion, blogs and social networks). Analyzing and observing these opinions enable individuals and organizations to make their decision easier. However, since there are a lot of websites on the Internet, it is difficult to monitor the huge volume of opinionated text manually. Moreover, it is also known that human analysis may have a bias caused by analyzer's subjectivity toward some services and products. To overcome these limitations, recent researchers have raised a lot of tasks in opinion mining area to build systems that are able to analyze opinionated text automatically.

Aspect-based sentiment analysis (ABSA) is known as a significant task in opinion mining. The task aims to extract explicit aspects of an entity, along with sentiment expressed towards these aspects. To achieve this goal, two subtasks are performed: aspect term extraction (ATE) and aspect polarity classification (APC). ATE is a task to extract aspects of a target (i.e. product or service) from a review text, while APC is a task to classify the polarity of the extracted aspects into the positive, negative or neutral. However, recent work has solved these two subtasks separately or has only focused on either subtask. In addition, a sequential model of two subtasks may cause chain errors from ATE to APC and designing and running two models consume too many resources. In this thesis, we introduce a new problem named aspect term extraction and polarity classification (ATEPC) that is able to deal with two subtasks at the same time. Then, we propose several deep learning models to address this problem.

We first formulate the ATEPC task by considering new output labels. While IOB encoding is used to indicate beginning, inside, and outside of an aspect term in the ATE task and a set of categories {positive, negative, neutral} indicating the polarity of an aspect is used in the APC task, ATEPC uses a new "polarity IOB encoding" to indicate not only aspect term but also its polarity. A set of IOB labels are defined as $\{B_{pos}, B_{neg}, B_{neu}, I_{pos}, I_{neg}, I_{neu}, O\}$. For example, B_{pos} indicates the beginning of an positive aspect term, I_{pos} indicates the inside of an positive aspect term, O indicates the outside of aspect term, etc. By using this IOB encoding, we perform ATEPC task as a sequential labeling problem to extract aspect terms and their polarity.

A new method called BiLSTM+CRF is proposed to solve the ATEPC problem. It is based on a combination of bidirectional long short-term memory networks (BiLSTM) and conditional random fields (CRF). After preprocessing and tokenising a review sentence, each word in the sentence is represented as a continuous vector called word embedding , which are pre-trained from corpora by existing methods. The model then feeds word embedding into bidirectional LSTM in order to learn abstract representation of words in the sentence for not only aspect terms but also their polarity. Finally, the results from BiLSTM are passed as the input of CRF to predict the output sequence for the input sentence.

Next, the attention mechanism is introduced to the model BiLSTM+CRF in order to improve the performance in the ATE and APC tasks. The basic idea of attention mechanism is that the model should focus on the suitable context in the sentence when it decides an output for an target (aspect term). For example, if the model considers the target “voice” in the sentence “The voice of my Moto phone was unclear,” the model should pay more attention to suitable context words such as “voice” and “unclear” rather than other unrelated words such as “the”, “was.” A requirement of the conventional attention mechanism is that it requires a target in order to incorporate new information. The target should be given to guess what words should be paid more attention. It can be applicable for a classification problem or generation problem. However, in a sequential labeling problem like ATEPC, no target is given to a model. Therefore, we propose a contextual attention mechanism that is able to overcome this problem. The basic idea of the contextual attention mechanism is that it treats each word in the sentence as a target and apply the attention mechanism for them. Using this technique, each word in the sentence is represented by not only the information of the word itself but also the information from its words in context. Thus the information of opinion words is directly incorporated into the aspect terms. Moreover, as for out-of-vocabulary (OOV) words, this technique also provides a better representation for them using information from context words instead of using random representation. To sum, the advantage of this attention mechanism is that it not only directly incorporate polarity information of terms in long distance but also synthesize word vectors of OOV words appropriately. These advantages play an important role in boosting the model performance in the APC task.

We carry out experiments to evaluate our proposed methods. Although we propose a joint model of the ATE and APC tasks, the performance of the models are evaluated separately. While the F1-measure is used as an evaluation criteria for ATE, the performance of APC is evaluated with respect to the accuracy. The accuracy of the APC task is defined as a proportion of the aspect terms whose polarity are correctly identified to the total number of the correctly extracted aspect terms. When a set of aspect terms correctly extracted by two methods are different, comparison of the accuracy of these methods is not completely fair. Therefore, for fair comparison of two models, we define micro-accuracy that evaluates the polarity classification of only aspect terms correctly extracted by both models.

Although BiLSTM+CRF is a common model, it performs well when it is applied to the ATEPC task. The experimental results on SemEval datasets show that BiLSTM+CRF outperforms several baseline models in the ATE task and gives a promising result in the APC task by dealing with ATE and APC at the same time. Moreover, the experiments also show that the model using the contextual attention mechanism (CATT+BiLSTM+CRF) have a significant improvement in the APC task compared with BiLSTM+CRF. Furthermore, we visualized the attention parameters trained by the model and confirmed that the contextual attention mechanism was appropriate to incorporate polarity information of words in long distance. In addition, an error analysis is carried out to reveal the major problems of the proposed models.

The contribution of this thesis is summarized as follows. Firstly, we proposed a new framework named ATEPC which could deal with two subtasks ATE and APC in opinion mining simultaneously. Secondly, we proposed the contextual attention mechanism to improve APC performance in ATEPC task. We confirmed that our proposed method outperformed previous work in the ATE task and achieved comparable results in the APC task via several experiments using SemEval datasets.

Acknowledgements

I would like to express the deepest appreciation to my supervisor, Associate Professor Kiyoaki Shirai who was always willing to give me the advice on the problem that I was stuck. He not only taught me the research skills but also helped me to improve the English skills, especially writing. Without his guidance, my thesis could not be finished on time. I also would like to thank Associate Professor Minh Le Nguyen for his support as a second supervisor. Moreover, I would like to thank Associate Professor Minh Le Nguyen, Associate Professor Shinobu Hasegawa, and Associate Professor Kokoro Ikeda for their useful comments in mid-term and final-term defenses that are key points in improving the quality of my thesis.

I would like to thank the lecturers from Vietnam National University - Ho Chi Minh City and JAIST for giving the essential knowledge that is really helpful in the research. I greatly appreciate the Collaborative Education Program organized by Japan Advanced Institute of Science and technology and Vietnam National University - Ho Chi Minh City, which gave me a chance to study abroad at JAIST.

Finally, special thanks to my family, all members of Shirai's and Hasegawa's laboratories, and my friends. They not only gave me the encouragement and inspiration but also was beside me whenever I was in trouble.

Contents

1 Introduction **6**

1.1 Background . . . 6

1.2 Goal . . . 7

1.3 Contributions . . . 7

1.4 Thesis Outline . . . 8

2 Related Work **9**

2.1 Sentiment Analysis . . . 9

2.1.1 Aspect Term Extraction (ATE) . . . 11

2.1.2 Aspect and Opinion Terms Co-extraction (AOTE) . . . 12

2.1.3 Aspect Polarity Classification (APC) . . . 13

2.2 Deep learning . . . 14

2.2.1 Word Embedding . . . 14

2.2.2 Recurrent Neural Network (RNN) . . . 15

2.2.3 Long Short-Term Memory (LSTM) . . . 16

2.2.4 Attention Mechanism . . . 18

2.3 Conditional Random Field (CRF) . . . 19

3 Proposed Method **22**

3.1 Task of Aspect Term Extraction and Polarity Classification (ATEPC) . . . 22

3.2 BiLSTM+CRF . . . 23

3.3 Contextual Attention Mechanism . . . 25

4 Experiments **30**

4.1 Data . . . 30

4.2 Evaluation Metric . . . 31

4.2.1 Evaluation criteria for ATE . . . 31

4.2.2 Evaluation criteria for APC . . . 32

4.3 Experimental Settings . . . 34

4.4 Effectiveness of Word Embeddings . . . 35

4.5 Evaluation of ATEPC models . . . 37

4.5.1 Results of ATE task . . . 37

4.5.2 Results of APC task . . . 39

4.6 Attention Visualization . . . 40

4.7 Error Analysis 40

5 Conclusions **43**

5.1 Summary 43

5.2 Future Work 44

This dissertation was prepared according to the curriculum for the Collaborative Education Program organized by Japan Advanced Institute of Science and Technology and Vietnam National University - Ho Chi Minh City.

List of Figures

2.1 Input and output of aspect term extraction task. 12
2.2 Input and output of aspect and opinion terms co-extraction task. 12
2.3 Input and output of aspect polarity classification task. 13
2.4 A visualization about male-female, verb tense and country-capital relationships between words in vector space 15
2.5 RNN with three time steps 16
2.6 LSTM with three time steps 17
2.7 Attention mechanism 19
2.8 Relation between Naive Bayes, logistic regression, HMM, and CRF [35] . . 20

3.1 BiLSTM+CRF architecture 23
3.2 How to incorporate polarity at distance position in the contextual attention mechanism 27
3.3 How to handle OOV words in the contextual attention mechanism 27
3.4 CATT+BiLSTM+CRF architecture 29

4.1 Converting output labels from ATEPC to ATE and APC 32
4.2 F-measure of the ATE task of BiLSTM+CRF with different word embeddings 35
4.3 F-measure of the ATE task of CATT+BiLSTM+CRF with different word embeddings 36
4.4 Attention visualization for some examples 41

List of Tables

4.1 SemEval datasets 31
4.2 Optimization of hyperparameters 34
4.3 F1 measure of ATE performance 38
4.4 Precision, recall, and F1-measure of ATE performance 39
4.5 Accuracy of APC performance 39
4.6 Micro-accuracy of BiLSTM+CRF and CATT+BiLSTM+CRF 40
4.7 Examples of results obtained by BiLSTM+CRF and CATT+BiLSTM+CRF 42

Chapter 1

Introduction

In this chapter, we first explain a background of our research in Section 1.1. Section 1.2 describes the motivation and goal of this work. Then we describe the contributions of this thesis in Section 1.3. Finally, the structure of the thesis is given in Section 1.4.

1.1 Background

Sentiment analysis or opinion mining is a field of analyzing opinionated text in natural language processing. It aims to understand people's opinions, appraisals, attitudes, and emotions toward entities, individuals, issues, events, topics and their attributes [19]. The task is very useful to not only individuals but also organizations in their decision making. For example, potential customers can know advantages and drawbacks of a service before using it by reading its reviews from other users. Enterprises always want to know the customer's opinions to improve the quality of their product in the next version.

With the quick development of social media (e.g. social networking service such as Facebook [1], microblogging service such as Twitter [2], forum such as reddit [3], and so on), people can easily express feeling or their opinion about a product or a service. This leads to difficulty in monitoring and analyzing such kinds of information manually because there is a huge amount of opinionated text on the social media. In addition, it is also known that human analysis of text information is subject to considerable biases, e.g., people often pay greater attention to opinions that are consistent with their own preferences [19]. Therefore, automated opinion mining and summarization systems are needed. Instead of using manual methods, researchers become more interested in analyzing opinionated text automatically. These systems can overcome the limitation of subjective opinion bias and shortage of opinionated text that can be observed by human.

To analyze opinions in the Internet automatically, researchers proposed a lot of tasks in opinion mining such as sentence-level sentiment analysis, document-level sentiment analysis, aspect-based sentiment analysis (ABSA), text classification, etc. Among these

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

[2] https://twitter.com

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

tasks, ABSA is more sophisticated and requires a lot of knowledge. Unlike sentence-level or document-level sentiment analysis, where overall sentiment in a sentence or document is automatically determined, ABSA considers aspects of a product that people are talking or expressing their opinions about. For a given sentence, ABSA aims at extracting explicit aspects of an entity along with the sentiments expressed towards these aspects. For example, from the sentence "The voice of my Moto phone was unclear, but the camera was good," it is desired to extract the aspect terms "voice" and "camera" along with "negative" and "positive" sentiment, respectively. To achieve this goal, two subtasks are performed: aspect term extraction (ATE) and aspect polarity classification (APC). ATE aims to extract the aspect terms and then APC aims to determine the polarity expressed towards these aspect terms.

1.2 Goal

To the best of our knowledge, while ATE and APC subtasks were defined as the first steps of opinion mining development [18], they were regarded as independent subtasks. Recent researchers have dealt with two subtasks, ATE and APC, separately or they have only focused on either subtask. However, to perform the APC task on real-life data, we must first perform ATE to identify the aspect terms to be classified with respect to the polarity. This is a kind of sequential task that consumes a lot of resources for implementation and running of two models.

Moreover, the sequential model of two subtasks may cause chain errors from ATE to APC. Let us consider an example sentence "The voice of my Moto phone was unclear, but the camera was good", if the aspect term "camera" is not extracted by ATE model, the APC model cannot perform further even when it has ability to classify the polarity correctly. Note that the polarity of "camera" can be easily identified because "camera was good" is a simple and typical sentence to express positive opinion. Such chain errors might be more serious when the ATE modules is designed to perform high precision but low recall. In addition, useful features for ATE and APC may be overlapped and related to each other such as word embedding, part-of-speech feature, sentiment score feature, etc.

The goal of this study is to propose a joint model that is able to deal with two subtasks ATE and APC simultaneously. It is motivated by the above discussion, that is, handing ATE and APC at the same time is worth to be investigated. We will explore a task definition of a joint problem of ATE and APC, an appropriate method based on deep learning, and a possible improvement of the model. Furthermore, we will conduct an experiment to evaluated our proposed methods.

1.3 Contributions

The contribution of this thesis is as follows.

- We propose a new framework, named ATEPC, which can simultaneously deal with two subtasks, ATE and APC, in opinion mining.

- We propose a contextual attention mechanism to improve APC performance in ATEPC task. It aims at two improvements: (1) it enables the model to directly incorporate polarity information of opinion terms at long distance into an aspect term, (2) it provides better representation of out-of-vocabulary (OOV) word by considering words in the context.

- We conduct experiments on SemEval datasets to show that our proposed model produce better results in ATE and gives promising results in APC by dealing with ATE and APC at the same time.

1.4 Thesis Outline

The rest of this thesis is organized as follows.

- In chapter 2, we describe the related work about sentiment analysis, deep learning, and recent work on the ATE and APC tasks.

- In chapter 3, we illustrate the problem of aspect term extraction and polarity classification (ATEPC), and a model using bidirectional long short-term memory networks (BiLSTM) combined with conditional random field (CRF). In addition, we also describe a contextual attention mechanism, a powerful technique to boost APC performance in the ATEPC task.

- In chapter 4, we show experimental results on different datasets of SemEval. Furthermore, we also show an error analysis on the ATE and APC task.

- In chapter 5, we summarize the contribution of this study and some future work for the ATEPC task and the implemented models.

Chapter 2

Related Work

This chapter consists of three sections. In Section 2.1, we first introduce what sentiment analysis is. Then we describe literature reviews of related work in sentiment analysis including aspect term extraction (ATE), aspect polarity classification (APC), and aspect and opinion terms co-extraction (AOTE). Section 2.2 introduces several related deep learning models that we use to learn the abstract representation for words in the sentence. Finally, Section 2.3 presents a powerful sequential labeling model called conditional random field (CRF), since it is used for extraction of aspects and identification of their polarity in this study.

2.1 Sentiment Analysis

According to Liu et al. [19], opinions can be expressed about anything, e.g., a product, a service, an individual, an organization, an event, or a topic, by any person or organization. An opinion can be represented as a quintuple, $(e_i, a_{ij}, oo_{ijkl}, h_k, t_l)$ where e_i is the name of an entity, a_{ij} is an aspect of e_i, oo_{ijkl} is the orientation of the opinion about aspect a_{ij} of entity e_i, h_k is the opinion holder, and t_l is the time when the opinion is expressed by h_k. The opinion orientation oo_{ijkl} can be positive, negative or neutral, or be expressed with different strength/intensity levels. When an opinion is not toward any particular aspect, we denote it by using special aspect GENERAL. Liu et al. showed an illustrative example of quintuples as follows [19].

> **Example 1**: (Blog Posting) Posted by: bigXyz on Nov-4-2010: (1) I bought a Motorola phone and my girlfriend bought a Nokia phone yesterday. (2) We called each other when we got home. (3) The voice of my Moto phone was unclear, but the camera was good. (4) My girlfriend was quite happy with her phone, and its sound quality. (5) I want a phone with good voice quality. (6) So I probably will not keep it.

The following four quintuples are supposed to be extracted from the above example sentences.

- (Motorola, voice_quality, negative, bigXyz, Nov-4-2010)
- (Motorola, camera, positive, bigXyz, Nov-4-2010)
- (Nokia, GENERAL, positive, bigXyz's_girlfriend, Nov-4-2010)
- (Nokia, voice_quality, positive, bigXyz's_girlfriend, Nov-4-2010)

We can easily recognise that each quintuple describes orderly: entity, aspect, sentiment, opinion holder, and time corresponding to the definition of $e_i, a_{ij}, oo_{ijkl}, h_k, t_l$ respectively.

To extract quintuples from a text, we have to perform the following tasks:

- **Task 1: Entity extraction and grouping**

  In this task, a system extracts all entities in an opinionated document D and groups synonymous entities into entity clusters. Each cluster indicates an unique entity e_i. For example, a system extracts three entities "Motorola", "Nokia", and "Moto" and then generates two entity clusters "Motorola" and "Nokia", in which "Motorola" and "Moto" are merged into a single cluster "Motorola".

- **Task 2: Aspect extraction and grouping**

  In this task, a system extracts all aspects and groups of synonymous aspects into aspect clusters. Each aspect cluster a_{ij} indicates a particular aspect of e_i. For example, three aspects "camera", "voice quality", and "sound" are extracted in Example 1, and group "voice" and "sound" together as these synonymous aspects represent the same aspect.

- **Task 3: Opinion holder and time extraction**

  In this task, a system extracts the name of opinion holder and time in document D. From Example 1, "bigXyz" are "bigXyz's_girlfriend"are extracted as opinion holders, while "Nov-4-2010" is extracted as time.

- **Task 4: Aspect sentiment classification**

  In this task, a system determines the polarity for each opinion on an aspect of an entity such as positive, negative or neutral. In Example 1, the polarity of "voice_quality" of "Motorola" is classified as negative, while rest of the aspect or product is classified as positive.

- **Task 5: Opinion quintuple generation**:

  In this task, a system generates all quintuples $(e_i, a_{ij}, oo_{ijkl}, h_k, t_l)$ as a final result like 4 quintuples in Example 1.

In this work, we mainly focus on aspect extraction (task 2) and aspect sentiment classification (task 4). It has also been known as aspect term extraction (ATE) and aspect polarity classification (APC) respectively.

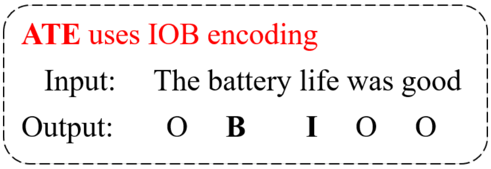
2.1.1 Aspect Term Extraction (ATE)

Recent researches on aspect term extraction mainly work on online reviews. It aims to extract aspect terms in a sentence or a document. For example, in a review of the sentence “The voice of my Moto phone was unclear, but the camera was good,” it is desired to extract the aspect terms “voice” and “camera” respectively. Although there are a lot of existing methods, we can divide into two categories: unsupervised methods and supervised methods.

As for unsupervised methods, Hu et al. first introduced a rule-based approach that consists of 2 steps to collect aspects [11]. Firstly, frequent nouns and noun phrases were collected as aspect terms. This is due to the idea that the vocabulary usually converges when people comment on different aspects of a product. Secondly, they tried to collect infrequent nouns and noun phrases by exploiting the relationships or the cooccurrence between aspects and opinion words. For instance, by exploiting the following sentence “The pictures are absolutely amazing”, which contains “pictures” as an aspect term and “amazing” as an opinion term, we can easily extract an aspect“software” from another sentence “The software is amazing.” Because a term near an opinion word (“amazing” in this case) tends to be an aspect term. Popescu et al. improved step 1 by trying to remove noun phrases that may not be product aspects/features such as “picture of”, “picture comes with”, and so on [29]. He et al. proposed a neural based model that can discover and extract coherent aspects by exploiting the distribution of word co-occurrences through the use of neural word embedding [9].

Several methods based on topic modeling such as Latent Dirichlet Allocation (LDA) are also considered as unsupervised learning methods. Titov and McDonald proposed multi-grain topic models which were based on extensions of standard topic modeling methods such as LDA and probabilistic latent semantic analysis (PLSA) to discover local rateable aspects [38]. These models not only extracted aspects but also grouped the words indicating the same or related aspects together. For example, “volume” and “excellent” are a part of the same topic “sound quality” in a speaker domain. However, their models did not separate the aspect and opinion words. Therefore, join topic-sentiment model, positive sentiment model, and negative sentiment model were proposed to separate these aspect and opinion words [17,21]. Zhao et al. also proposed a MaxEnt-LDA hybrid model to discover syntactic features between aspect and opinion words [44]. These features helped the model to separate aspect and opinion words easily.

With the rapid development of machine learning, researchers are more interested in supervised learning methods by treating ATE as a sequential labeling problem. In this problem, IOB encoding is often used to indicate beginning, inside, and outside of an aspect term. Figure 2.1 illustrates how IOB encoding is used to define the input and output of this task. Note that the output in Figure 2.1 means that “battery life” is extracted as an aspect. Yin et al. proposed an approach to learn distributed representations of words and dependency paths by unsupervised learning [43]. They then fed the learned embedding of the words and dependency paths along with some hand-crafted features into the conditional random field. Poria et al. used a 7-layer deep convolutional neural network and proposed a set of linguistic patterns as a post-processing to tag each word

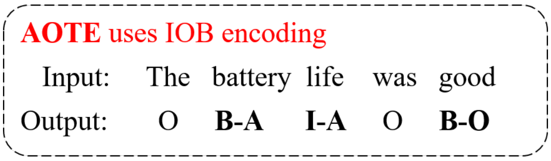


ATE uses IOB encoding

Input:	The	battery	life	was	good
Output:	O	**B**	**I**	O	O

"B" indicates beginning of the aspect term, "I" indicates inside of the aspect term, and "O" indicates outside.

Figure 2.1: Input and output of aspect term extraction task.



AOTE uses IOB encoding

Input:	The	battery	life	was	good
Output:	O	**B-A**	**I-A**	O	**B-O**

"B-A" indicates begin of the aspect term, "I-A" indicates inside of the aspect term, "B-O" indicates begin of the opinion term, "I-O" indicates inside of the opinion term, and "O" indicates outside.

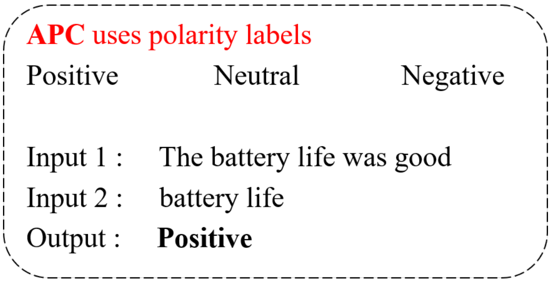
Figure 2.2: Input and output of aspect and opinion terms co-extraction task.

in opinionated sentences as either aspect or non-aspect word [30]. Xu et al. also used a deep convolutional neural network employing two types of pre-trained embeddings for aspect extraction: general word embeddings and domain-specific word embeddings [42]. Their model achieved a state-of-the-art result on SemEval datasets.

2.1.2 Aspect and Opinion Terms Co-extraction (AOTE)

The task of aspect and opinion terms co-extraction (AOTE) aims to extract not only aspect terms but also opinion terms in a sentence or a document. This task is related to or a variation of the aspect term extraction (ATE) task. For example, in the review sentence "The voice of my Moto phone was unclear, but the camera was good," it is desired to extract "voice" and "camera" as aspect terms along with "unclear" and "good" as opinion terms respectively. Figure 2.2 also illustrates how IOB encoding is used to define the input and output of this task. Note that the output in Figure 2.2 means that "battery life" is extracted as an aspect and "good" is extracted as an opinion word.

The rule-based approach of Hu et al, which is described in Subsection 2.1.1 as a method of ATE task, is also considered as AOTE task. Wang et al. argued that the recursive neural network that was performed in a dependency tree could automatically learn the representation of phrases and words [41]. They fed hidden state representation of words with hand-crafted features into conditional random field (CRF) to perform AOTE. The results of their experiments showed that dual propagation of pairs of aspect and opinion terms were able to boost the performance of the model. Moreover, their model achieved the state-of-the-art result on the laptop dataset of SemEval 2014. However, since this



APC uses polarity labels

Positive	Neutral	Negative

Input 1 :	The battery life was good
Input 2 :	battery life
Output :	**Positive**

“Positive” indicates positive opinion for the aspect (input 2) in the sentence (input 1).

Figure 2.3: Input and output of aspect polarity classification task.

model has a problem that it heavily relies on results of syntactic parsing, another model with multi-layer attentions for AOTE task has been proposed [40]. This model achieved a state-of-the-art result on the restaurant dataset of SemEval 2014.

2.1.3 Aspect Polarity Classification (APC)

In the APC task, the relationship between the target (i.e. aspect term) and the words in the context is considered to classify the polarity of the target, i.e. positive or negative. Therefore, the term “target-dependent sentiment classification (TDSC)” usually refers to the APC task in recent work. Figure 2.3 illustrates the input and output of this task. In this task, both a sentence (Input 1) and an aspect (Input 2) are given as inputs, then the polarity of the aspect is obtained as an output.

Jiang et al. first emphasised the importance of targets by showing that 40% of sentiment analysis errors were caused by not considering them in classification [13]. They incorporated seven target-dependent features and other sentiment features into the Support Vector Machine (SVM), and subsequently obtained a significant improvement. Models for TDSC realised by a recursive neural network on dependency and constituent trees were proposed by Dong et al. [7] and Nguyen and Shirai [24]. The recursive neural network model composed an orderly abstract representation of the context words, from farthest to nearest of the target in the tree. It then produced a representation of the target at the final step. This means that if a sentence contains multiple targets, then each one at a different position has its own way to incorporate information from its context words in the tree.

Instead of using syntactic relations between a target and context words as used in the previous work, Tang et al. used two long short-term memory (LSTM) networks running towards to the target from left and right, respectively [36]. The final states of both networks were concatenated to predict the polarity of the target. Tang et al. claimed that the importance of each context word was different when inferring sentiment polarity of a target [37]. They implemented this idea using a memory network that could capture the importance of context words via an attention mechanism. Chen et al. also used the same idea of the importance of context words and proposed multiple attention mechanisms to capture sentiment features isolated in a long distance [3]. Their proposed model achieved state-of-the-art results.

2.2 Deep learning

Recently, deep learning has dramatically developed in different fields of computer science, especially natural language processing. In natural language processing tasks such as sentiment analysis, machine translation and text summarization, it has been proved that deep learning could achieve high performance without using any additional knowledge comparing to a lexicon-based method or a featured-based method [1, 31, 32]. One of the reasons is that deep learning is able to learn the latent structured representation (abstract representation) of unstructured data in vector space. Since our proposed method is based on deep learning, this section will describe several models and techniques of deep learning , especially ones that is related to our work.

2.2.1 Word Embedding

Word embedding is high-quality distributed vector representation that captures a large number of precise syntactic and semantic properties of words. It helps machine learning models to achieve better performance in natural language processing tasks by grouping similar words [22]. It means that synonymous words are allocated in adjacent location in vector space. The learned vectors explicitly encode many linguistic regularities and patterns. For example, vec("Madrid") - vec("Spain") + vec("France") is closer to vec("Paris") than to any other word vectors [22]. Allocation of a country and its capital in vector space is visualized in the right most figure in Figure 2.4. Figure 2.4 also visualizes the relations of male-female and verb tense (past form and progressive form) such as.

$$\text{vec}(\text{“king”}) - \text{vec}(\text{“man”}) + \text{vec}(\text{“woman”}) \approx \text{vec}(\text{“queen”})$$
$$\text{vec}(\text{“walking”}) - \text{vec}(\text{“walked”}) + \text{vec}(\text{“swam”}) \approx \text{vec}(\text{“swimming”}).$$

There are two popular methods to train word embedding in the natural language processing area: Word2Vec [1] and GloVe [2]. Both of them can capture the semantics of analogy very well. The difference between them is that Word2Vec preserves semantic analogies for basic arithmetic on the word vectors, e.g. vec("Madrid") - vec("Spain") + vec("France") is closer to vec("Paris"), while Glove preserves semantic analogies for global word-word co-occurrence statistics in a corpus. Recent work on natural language processing has shown that both are effective in most cases.

In this thesis, we investigate Word2Vec and GloVe as the input of the machine learning model. We also show the results of random word embedding to examine the effectiveness of Word2Vec and GloVe in our experiment.

[1] https://code.google.com/archive/p/word2vec/

[2] https://nlp.stanford.edu/projects/glove/

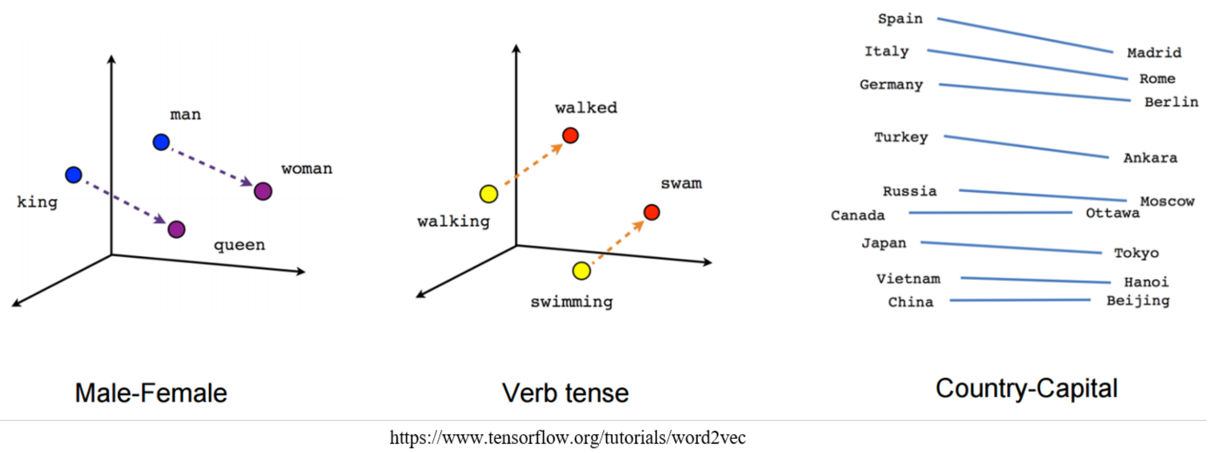


Figure 2.4: A visualization about male-female, verb tense and country-capital relationships between words in vector space

2.2.2 Recurrent Neural Network (RNN)

Recurrent neural network is a sequential learning model that is suitable for a temporal input such as a sentence and speech. It can be regarded as multiple copies of the same network, where each network passes information to a successor. Given an input sequence $X = [e_1, e_2, ..., e_N]$ where $e_t \in \mathbb{R}^{d_e}$ is d_e-dimensional input vector at time step t and N is the number of time steps, RNN performs recursively on each time step of the input as the following equation.

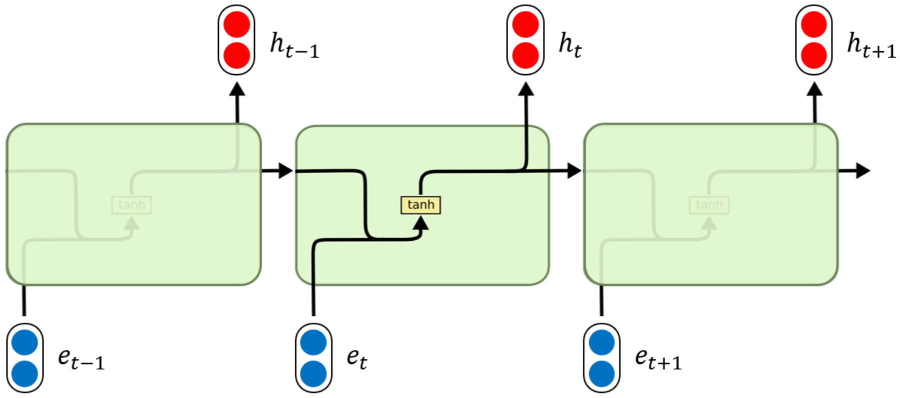
$$h_t = tanh(W_{rnn} \cdot [h_{t-1}, e_t] + b_{rnn}) \tag{2.1}$$

Parameters in this equation are explained as follows.

- $W_{rnn} \in \mathbb{R}^{d_h \times (d_h + d_e)}$ and $b_{rnn} \in \mathbb{R}^{d_h}$ are the model parameters which are updated and optimized by an optimization method such as stochastic gradient descent.
- $h_t \in \mathbb{R}^{d_h}$ is a hidden state of time step t. d_h is the number of hidden units in RNN. h_0 is initialized by zero vector.
- $tanh$ is the hyperbolic tangent, a kind of a non-linear function, that scales the output in a range from -1 to 1.
- square brackets indicate the concatenation operator.

Figure 2.5 is a graphical representation of this procedure.

At each time step, RNN considers not only current information e_t but also past information from previous time step h_{t-1} to create a vector representation for current state h_t. In this way, RNN can accept a sequential data as an input. As for classification problem like APC, we usually use the last hidden state h_N, which contains the whole information of the input, for classification. As for sequential labeling problems like ATE or AOTE, we



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

Figure 2.5: RNN with three time steps

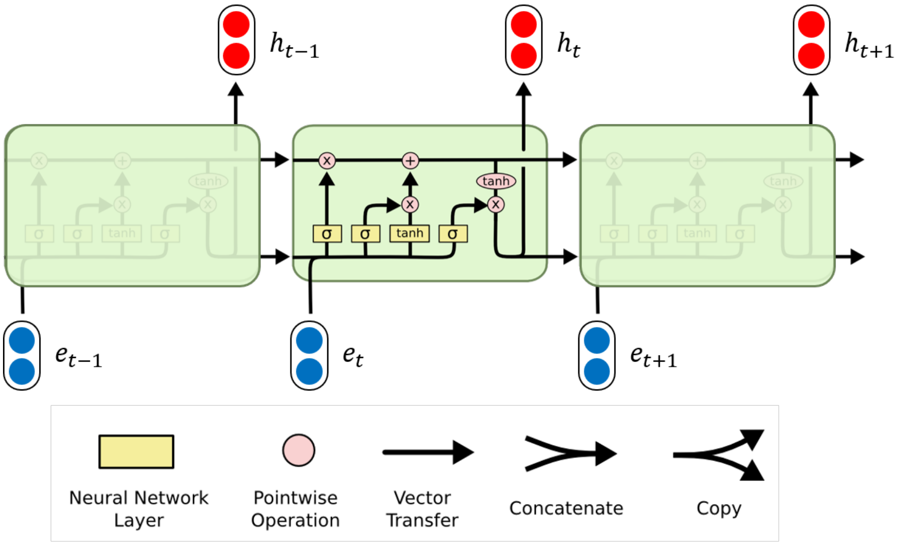
use the whole hidden state $[h_1, h_2, ...h_N]$ as output for further sequential labeling model such as a feed forward neural network or CRF.

Theoretically, RNN is able to capture long-term dependency by forwarding information at each time step. However, in practice, RNN still suffers from the problem that long dependency is not thoroughly considered [2]. Moreover, RNN also suffers from the vanishing gradient problem; the parameters in RNN are not updated in back propagation since a gradient is too small. It means the iterative learning of parameters of RNN is stopped too early [15]. As a result, the model can not obtain the optimised performance. To deal with these problems, long short-term memory is proposed.

2.2.3 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is an extended version of RNN. It was proposed by Hochreiter and Schmidhuber [10], and has been used popularly in different tasks in natural language processing, such as machine translation [4,34], named entity recognition [12,16], and sentiment analysis [3,40]. It is explicitly designed to use information at steps in long distance (steps that are far from the current step) by remembering past information for a long time.

LSTM also has a chain structure like RNN, but using a different way to incorporate information through time. Instead of using only one neural network layer, LSTM contains four layers (also called 4 gates) to automatically control the information, which is stored in the cell state through each time step. The following equations illustrate each recursive step in LSTM for a given input sequence $X = [e_1, e_2, ..., e_N]$ where $e_t \in \mathbb{R}^{d_e}$ is d_e-dimensional



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

Figure 2.6: LSTM with three time steps

input vector at time step t and N is the number of time steps.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2.2}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2.3}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{2.4}$$

$$\widetilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{2.5}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C}_t \tag{2.6}$$

$$h_t = o_t \odot tanh(C_t) \tag{2.7}$$

Parameters in these equations are explained as follows:

- f, i, o, and C stand for a forget gate, an input gate, an output gate, and a cell state respectively.
- $W \in \mathbb{R}^{d_h \times (d_h + d_e)}$ and $b \in \mathbb{R}^{d_h}$ are the model parameters which are updated and optimized by an optimization method such as stochastic gradient descent.
- $tanh$ is the hyperbolic tangent. σ is the sigmoid function. Both are non-linear functions. While $tanh$ scales the output in a range from -1 to 1, σ scales the output in a range from 0 to 1.
- $\odot$ represents an element-wise product.

Figure 2.6 is a graphical representation of this procedure.

The key element of LSTM is the cell state C which is used to store information through all time steps. LSTM has three gates named the forget gate f, input gate i, and output gate o to protect and control the cell state. The forget gate aims to decide what information the network should throw away from the cell state. The input gate decides the information to be newly stored in the cell state. The output gate decides what past information from the cell state and what current information from the input should go out to the hidden layer of the network.

Recently, LSTM is widely used for many natural language processing systems. One of the reasons is that LSTM is able to not only handle long-term dependency and the vanishing gradient problem but also achieve a good performance in a lot of tasks without using any hand-crafted features. In this work, we also use LSTM as a main part of the model to learn the abstract representation for each word in the sentence.

2.2.4 Attention Mechanism

Attention mechanism is a powerful technique in deep learning. The basic idea is that a model focuses on only several specific locations of data to incorporate information in past. This mechanism is very similar to visual attention mechanism found in human, which is the behavioral and cognitive process of selectively concentrating on a discrete aspect of information. Moreover, it also performs well and achieves a lot of state-of-the-art results in computer science such as computer vision [23], natural language processing [31], etc. Therefore, attention mechanism is a hot trend in deep learning.

Recent work on sentiment analysis also uses the attention mechanism to improve the model performance [3, 40]. In aspect polarity classification, the attention mechanism allows the model to pay more attention to the suitable context in the sentence when classify a polarity of a target (aspect term). For example, if the model considers the target "tech support" in the sentence "tech support would not fix the problem," the model should focus on suitable context words such as "not", "fix" and "problem". Given an input sequence $X = [e_1, e_2, ..., e_N]$ and a target vector e_{target}, the attention mechanism will perform as the following equations.

$$g_i = W_{Att_2}(tanh(W_{Att_1}[e_i, e_{target}] + b_{Att_1})) \tag{2.8}$$

$$\alpha_i = \frac{exp(g_i)}{\sum_j^N exp(g_j)} \tag{2.9}$$

$$u = \sum_{i=1}^{N} \alpha_i e_i \tag{2.10}$$

Parameters in these equations are explained as follows:

- $W_{Att_1} \in \mathbb{R}^{d_a \times 2*d_w}$, $b_{Att_1} \in \mathbb{R}^{d_a}$, and $W_{Att_2} \in \mathbb{R}^{1 \times d_a}$ are the parameters of the model. d_a is the number of units in this layer. These parameters play a role of deciding the amount of information from context words to be incorporated into the current word t.

- α_i is the amount of information from the word at the position i. Intuitively, when the vector of the target e_{target} and the vector of the i-th word e_i are similar, α_i becomes great.

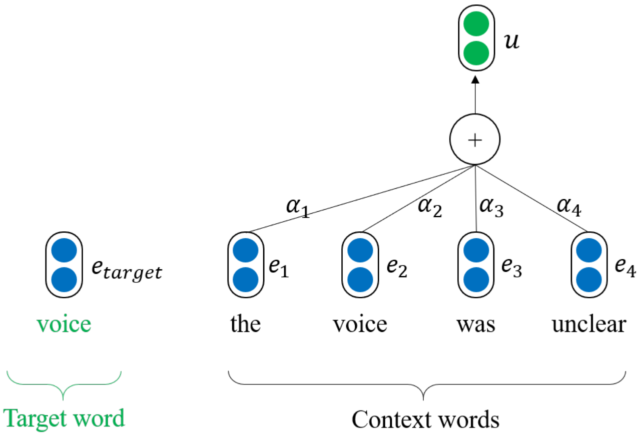


Figure 2.7: Attention mechanism

- u is a synthesis vector that is created by adding context word vectors with the weights α.

Figure 2.7 also describes this procedure. If the model considers the target word “voice”, it will pay more attention on suitable words such as “voice” and “unclear” rather than other words “the”, “was”. As a result, the amount of information from the words “voice” and “unclear” (α_2 and α_4), which are directly incorporated to u, will be larger than other words. As for aspect term polarity classification (APC), we usually use u to predict the sentiment of the target in the sentence because it contains major information of sentence such as “voice” and “unclear” as the previous description. As for the sequential labeling problem like ATE or AOTE, u is concatenated to e_i to provide a better representation including the contextual information of the target word.

2.3 Conditional Random Field (CRF)

Many natural language processing tasks such as part-of-speech tagging and named entity recognition can be formulated as a sequential labeling problem. The sequential labeling problem is a task to predict an output sequence $Y = [y_0, y_1, ..., y_N]$ of random variable for a given (observed) sequence of feature vectors $X = [x_0, x_1, ..., x_N]$. Each x_t consists of different information about the word at position t such as orthographic feature, prefix, suffix, binary feature in a lexicon, etc. Although we can use a graphical model like hidden Markov model (HMM) to handle the sequential labeling problem by modeling a joint probability distribution $p(Y, X)$, this approach contains several limitations [35]. Firstly, the dimensionality of x_t may be large. Secondly, the features have a complex dependency to construct a probability distribution among them. Finally, modeling the dependency among inputs can make it hard to train the model, but ignoring it makes the performance of the model worse.

A solution for these problems is to use conditional random field (CRF) that maximizes directly the conditional distribution $p(Y|X)$, which is also an objective function of

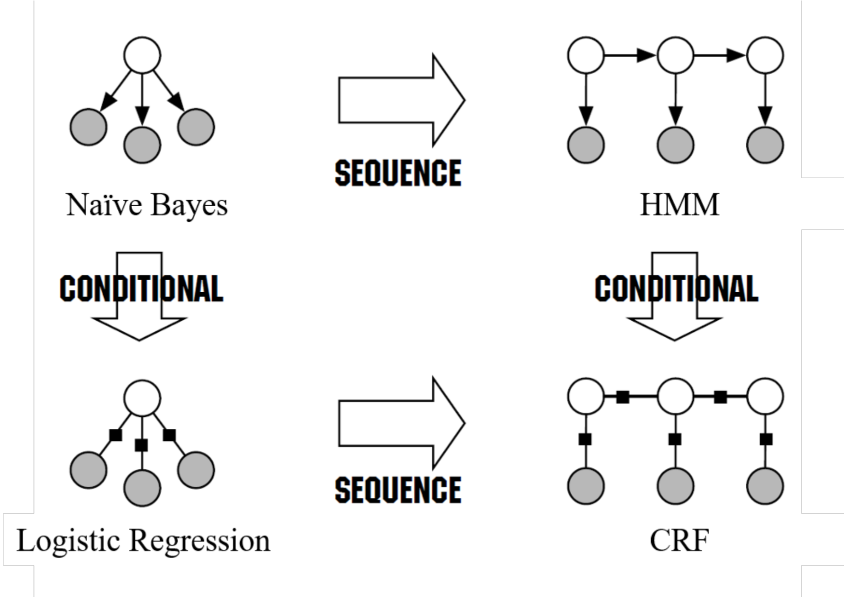


Figure 2.8: Relation between Naive Bayes, logistic regression, HMM, and CRF [35]

the classification problem. CRF takes the advantages for not only classification (logistic regression) but also graphical modeling, by combining the ability to observe high dimensional input feature with the ability to handle the relation between random variables y. Moreover, the dependency between input features do not affect to the conditional distribution $p(Y|X)$. Figure 2.8 describe the above advantages of CRF that are inherited from HMM and logistic regression model. In this figure, "CONDITIONAL" means that the model B is derived from the model A, where the original model A uses a joint probability distribution $p(Y,X)$ as an objective function while the model B maximizes a conditional probability $p(Y|X)$. On the other hand, "SEQUENCE" means that the model B is an extension of the model A, where the model A accepts and classify a single input while the model B can accepts and classify a sequence.

Now we address the details of CRF. Let us suppose that X and Y are a pair of input and output sequence. They are denoted as $X = [x_0, x_1, ..., x_N]$ and $Y = [y_0, y_1, ..., y_N]$ respectively. $x_t \in \mathbb{R}^{d_e}$ is d_e-dimensional feature vector at time step t and N is the number of time steps. Conditional random field calculates a distribution $p(Y|X)$ in the Equation (2.13).

$$P = X^T W_{crf} + b_{crf} \tag{2.11}$$

$$score(X,Y) = \sum_{t=1}^{N-1} A_{y_t,y_{t+1}} + \sum_{t=1}^{N} P_{t,y_t} \tag{2.12}$$

$$p(Y|X) = \frac{e^{score(X,Y)}}{\sum_{\widetilde{Y} \in Y_X} e^{score(X,\widetilde{Y})}} \tag{2.13}$$

Parameters in these equations are explained as follows.

- $X \in \mathbb{R}^{d_e \times N}$ is the matrix of all feature vectors.

- $P \in \mathbb{R}^{N \times K}$ is the output score matrix of K output tags. $P_{t,j}$ corresponds to the score of the j^{th} tag of the t^{th} feature vector x_t.

- $A \in \mathbb{R}^{(K) \times (K)}$ is a matrix of transition scores. $A_{i,j}$ is $p(y_j|y_i)$, which is the conditional probability of transition from tag i to tag j.

- A, $W_{crf} \in \mathbb{R}^{d_e \times K}$ and $b_{crf} \in \mathbb{R}^K$ are the model parameters which are updated and optimized by an optimization method such as stochastic gradient descent.

- Y_X presents all possible output sequences for an input sequence X.

In training phase, we maximize $p(Y|X)$ as an objective function to update model parameters. In test phase, we choose a sequence $y*$ that obtains the best score function as in Equation (2.14).

$$y^* = \underset{\widetilde{Y} \in Y_X}{\operatorname{argmax}} \, score(X, \widetilde{Y}) \tag{2.14}$$

Chapter 3

Proposed Method

In this chapter, we first present the problem of solving ATE and APC simultaneously, which is named aspect term extraction and polarity classification (ATEPC). Then we describe a BiLSTM+CRF model that solves both subtasks at the same time. This is based on a combination of bidirectional LSTM networks (BiLSTM) and conditional random field (CRF), which is similar to neural-based sequential labeling models for other tasks [5, 12, 16]. Finally, we propose a contextual attention mechanism to boost the APC performance in the ATEPC task.

3.1 Task of Aspect Term Extraction and Polarity Classification (ATEPC)

The input and output of the ATEPC task are defined as follows. Similarly to named entity recognition, the sequential labeling with IOB encoding is used for ATEPC. Each sentence is represented by a pair $<S, Y>$, where S and Y stand for the input and output, respectively. $S = [s_1, s_2, ..., s_n]$ is a list of words in the sentence, while $Y = [y_1, y_2, ..., y_N](y_t \in \{B_{pos}, B_{neu}, B_{neg}, I_{pos}, I_{neu}, I_{neg}, O\})$ is a list of corresponding output labels for each word in S. B, I, O indicate beginning, inside, and outside of the aspect term, respectively. The suffix of the label B and I indicates the polarity of the aspect, where *pos*, *neu*, and *neg* represent positive, neutral, and negative, respectively. For instance, the sentence S “The voice was unclear but the battery life was good” should be labeled as the output $Y = [O, B_{neg}, O, O, O, O, B_{pos}, I_{pos}, O, O]$. It means that there are two aspect terms: “voice” and “battery life”. It also means that the author of this sentence expresses a negative opinion for “voice” and a positive opinion for “battery life”. Opinion terms that represent the polarity for the aspect terms such as “unclear” and “good” in these example are classified as non-polarity-related class, i.e. O in our definition of ATEPC task. It means that both polarity words and non-polarity words are classified as O and not distinguished each other. It might be better to distinguish them by classifying the opinion words as O_{pos} or O_{neg}. However, we do not choose this approach because it requires more training data due to increase of the classification labels. We believe that a deep learning model, such as bidirectional LSTM, is able to incorporate polarity

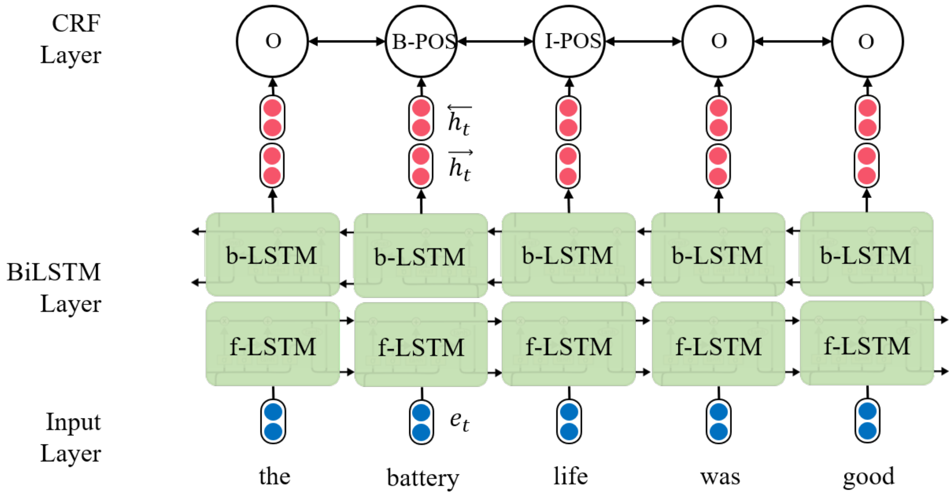


Figure 3.1: BiLSTM+CRF architecture

information into classification of aspect terms at distant positions. Because LSTM can consider long dependencies in a sentence in general.

3.2 BiLSTM+CRF

Recently, BiLSTM+CRF has been known as the most powerful sequence labeling model for NER [6] without using any additional hand-crafted features. It is due to the fact that it inherits the advantage of not only learning representation by BiLSTM but also labeling the output sequence by CRF. Besides that, BiLSTM is able to deal with long-term dependency (opinion terms in the long distance) which is a major factor on improving APC performance. Therefore, it is worth to be investigated as a model for the ATEPC task.

The architecture of BiLSTM+CRF is illustrated in Figure 3.1. After preprocessing and tokenizing a review sentence $S = [s_1, s_2, ..., s_N]$, each word in a sentence is represented as a d_e-dimensional vector $e_i \in \mathbb{R}^{d_e}$ that is obtained from a word embedding matrix $E \in \mathbb{R}^{d_e \times |V_e|}$, where $|V_e|$ is a vocabulary size. Then the model feeds word embedding $(X = [e_1, e_2, ..., e_N])$ into bidirectional LSTM in order to learn abstract representation of words in the sentence for not only aspect terms but also their polarity.

As we mentioned in Subsection 2.2.3, LSTM performs from left to right of the sentence by incorporating past information at time $t - i$ into the information at the current time t. The information from the future context (from right to left of the sentence) may be also useful to get an abstract representation at time step t. Therefore, we run a second LSTM in the reverse direction where the input is $X' = [e'_N, e'_{N-1}, ..., e'_1]$. A pair of forward and backward LSTMs is called bidirectional LSTM [8]. The following equations illustrate each recursive step in BiLSTM.

$$\begin{align}
\overrightarrow{f_t} &= \sigma(\overrightarrow{W_f} \cdot [\overrightarrow{h_{t-1}}, x_t] + \overrightarrow{b_f}) \tag{3.1}\\
\overrightarrow{i_t} &= \sigma(\overrightarrow{W_i} \cdot [\overrightarrow{h_{t-1}}, x_t] + \overrightarrow{b_i}) \tag{3.2}\\
\overrightarrow{o_t} &= \sigma(\overrightarrow{W_o} \cdot [\overrightarrow{h_{t-1}}, x_t] + \overrightarrow{b_o}) \tag{3.3}\\
\overrightarrow{\tilde{C}_t} &= tanh(\overrightarrow{W_C} \cdot [\overrightarrow{h_{t-1}}, x_t] + \overrightarrow{b_C}) \tag{3.4}\\
\overrightarrow{C_t} &= \overrightarrow{f_t} \odot \overrightarrow{C_{t-1}} + \overrightarrow{i_t} \odot \overrightarrow{\tilde{C}_t} \tag{3.5}\\
\overrightarrow{h_t} &= \overrightarrow{o_t} \odot tanh(\overrightarrow{C_t}) \tag{3.6}
\end{align}$$

$$\begin{align}
\overleftarrow{f_t} &= \sigma(\overleftarrow{W_f} \cdot [\overleftarrow{h_{t-1}}, x'_t] + \overleftarrow{b_f}) \tag{3.7}\\
\overleftarrow{i_t} &= \sigma(\overleftarrow{W_i} \cdot [\overleftarrow{h_{t-1}}, x'_t] + \overleftarrow{b_i}) \tag{3.8}\\
\overleftarrow{o_t} &= \sigma(\overleftarrow{W_o} \cdot [\overleftarrow{h_{t-1}}, x'_t] + \overleftarrow{b_o}) \tag{3.9}\\
\overleftarrow{\tilde{C}_t} &= tanh(\overleftarrow{W_C} \cdot [\overleftarrow{h_{t-1}}, x'_t] + \overleftarrow{b_C}) \tag{3.10}\\
\overleftarrow{C_t} &= \overleftarrow{f_t} \odot \overleftarrow{C_{t-1}} + \overleftarrow{i_t} \odot \overleftarrow{\tilde{C}_t} \tag{3.11}\\
\overleftarrow{h_t} &= \overleftarrow{o_t} \odot tanh(\overleftarrow{C_t}) \tag{3.12}
\end{align}$$

Parameters in these equations are explained as follows.

- $\overrightarrow{W} \in \mathbb{R}^{d_h \times (d_h + d_e)}$ and $\overrightarrow{b} \in \mathbb{R}^{d_h}$ are the model parameters of forward LSTM, while $\overleftarrow{W} \in \mathbb{R}^{d_h \times (d_h + d_e)}$ and $\overleftarrow{b} \in \mathbb{R}^{d_h}$ are the model parameters of backward LSTM. d_h is the number of hidden units in LSTM.
- The suffix of W and b indicates the gate which these parameters are belong to: f, i, o, C are a forget gate, an input gate, an output gate, and a cell state respectively.
- $\overrightarrow{h_t} \in \mathbb{R}^{d_h}$ is the current hidden state of the forward LSTM while $\overleftarrow{h_t} \in \mathbb{R}^{d_h}$ is current hidden state of the backward LSTM.

The output hidden states from bidirectional LSTM are concatenated as $h_t = [\overrightarrow{h}_t; \overleftarrow{h}_t]$, which is used as a representation of the word s_i. Finally, we transform the hidden state of words into output label space using an additional fully connected neural layer described in Equation (3.13).

$$P = H^T \cdot W_{nn} + b_{nn} \tag{3.13}$$

$H \in \mathbb{R}^{2*d_h \times N}$ is the matrix of all hidden states. $P \in \mathbb{R}^{N \times K}$ is the output score matrix of K tags. $P_{t,j}$ corresponds to the score of the j^{th} tag of the t^{th} word in the sentence. $W_{nn} \in \mathbb{R}^{2*d_h \times K}$ and $b_{nn} \in \mathbb{R}^K$ are the model parameters of this layer.

The CRF layer is similar to the model proposed by Lample et al. [16]. We define the prediction score as Equation (3.14) and softmax function over all possible tag sequences as Equation (3.15) for a sequence $Y = [y_1, y_2, ..., y_N]$.

$$score(X, Y) = \sum_{t=1}^{N-1} A_{y_t, y_{t+1}} + \sum_{t=1}^{N} P_{t, y_t} \tag{3.14}$$

$$p(Y|X) = \frac{e^{score(X,Y)}}{\sum_{\widetilde{Y} \in Y_X} e^{score(X, \widetilde{Y})}} \tag{3.15}$$

$A \in \mathbb{R}^{K \times K}$ is a matrix of transition scores. $A_{i,j}$ indicates a score of a transition from a tag i to a tag j. Y_X presents all possible tag sequences for an input sequence X.

During the training phase, we minimize the loss function as in Equation (3.16) and Equation (3.17).

$$loss(\theta) = -log(p(Y|X)) \tag{3.16}$$

$$log(p(Y|X)) = score(X, Y) - log(\sum_{\widetilde{Y} \in Y_X} e^{score(X, \widetilde{Y})}) \tag{3.17}$$

$\theta = \{\overleftarrow{W}, \overleftarrow{b}, \overrightarrow{W}, \overrightarrow{b}, W_{nn}, b_{nn}, A\}$ are the model parameters. Whereas $log(p(Y|X))$ is the log-probability of the correct tag sequence.

In the test phase, an output sequence of a given sentence is chosen as one that maximises the score function as in Equation (3.18).

$$y^* = \operatorname*{argmax}_{\widetilde{Y} \in Y_X} score(X, \widetilde{Y}) \tag{3.18}$$

3.3 Contextual Attention Mechanism

One of the most issues in the BiLSTM+CRF model is that it prefers maximising the performance of the ATE task to the APC task. This is due to the reason that ATEPC is more similar to ATE task than APC because ATEPC and ATE are sequential labeling problems. Moreover, LSTM does not explicitly but implicitly consider polarity information of opinion terms at long distance in the memory cell. This characteristic might be inappropriate for the APC task. For example, in the sentence "tech support would not fix the problem," BiLSTM+CRF model may be able to correctly extract the aspect "tech support", but often fails to classify the polarity of it. Although "not fix the problem" indicates the negative polarity of "tech support", the information of this phrase often fails to be transferred to the aspect term. Furthermore, the model uses random vectors to represent out-of-vocabulary (OOV) words (the words that not exist in the vocabulary) such as "excellenttt" and "b-a-d". Intuitively, random vectors are inappropriate as representation of words. Therefore, we propose to use an attention mechanism in order to automatically create the contextual information for each word in the sentence as well as produce better representation of OOV words as an input of BiLSTM+CRF model.

In the aspect polarity classification, the attention mechanism allows the model to pay more attention to the suitable context in the sentence when a polarity of a target (aspect term) is classified. For example, in the sentence “tech support would not fix the problem,” the most important words are “tech support”. It should be specified as a target in the attention mechanism. However, in an essential sequence labeling problem like ATEPC, the target (the aspect term in ATEPC) is unknown. Therefore, we propose a contextual attention mechanism that enable us to incorporate attention mechanism into BiLSTM+CRF without giving the target explicitly. It can not only consider the polarity information of opinion terms at long distance directly but also handle out-of-vocabluary words appropriately. These advantages play an important role in boosting the model performance of the polarity classification in the ATEPC task.

As we mentioned in the Subsection 2.2.4, a traditional attention model required an embedding of a target word e_{target} and a list of embedding of context words $[e_1, e_2, ..., e_N]$ to incorporate a new vector u, which is a vector representation of words in the context of the target word. In the contextual attention mechanism, we treat each context word as a target and apply the attention mechanism for them. The detail procedures of the contextual attention mechanism are shown as in Equation (3.19), (3.20) and (3.21).

$$g_i^t = W_{Att_2}(tanh(W_{Att_1}[e_i, e_t] + b_{Att_1})) \tag{3.19}$$

$$\alpha_i^t = \frac{exp(g_i^t)}{\sum_j^T exp(g_j^t)} \tag{3.20}$$

$$u_t = \sum_{i=1}^{T} \alpha_i^t e_i \tag{3.21}$$

Parameters in these equations are explained as follows.

- $W_{Att_1} \in \mathbb{R}^{d_a \times 2*d_w}$, $b_{Att_1} \in \mathbb{R}^{d_a}$, and $W_{Att_2} \in \mathbb{R}^{1\times d_a}$ are the parameters of the model. d_a is the number of units in this layer. These parameters play a role of deciding the amount of information from context words to be incorporated into the current word t.

- α_i^t is the amount of information from word i, which is used to incorporate into the current word t. N stands for the number of words in the sentence.

- u_t is the contextual attention information for word t in the sentence.

With this contextual attention mechanism, each word in the sentence is represented using not only the information itself but also the information from its context words. Thus the information of the opinion words at long distance is directly referred for the classification of the aspect term. In addition, it can provide better vector representation for OOV words. Figure 3.3 and 3.2 describe two of above situations in the sentence “it is fast and simpple to use.” Figure 3.2 shows how the attention mechanism works for the classification of the aspect term “use”. Note that the polarity of the opinion word “fast” in the context can be directly referred by estimating α_3^7 as a large value.. On

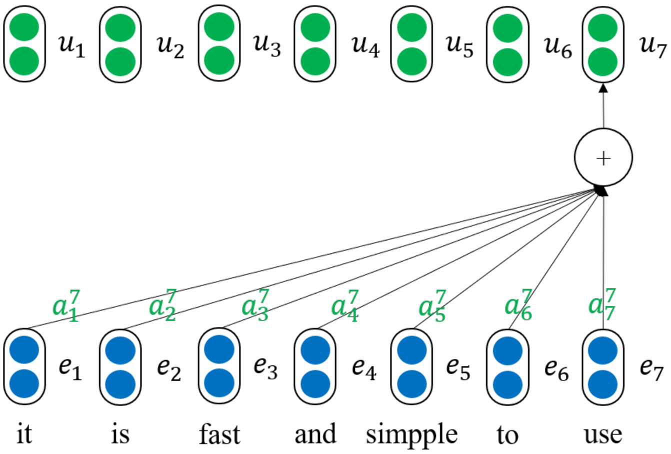


Figure 3.2: How to incorporate polarity at distance position in the contextual attention mechanism

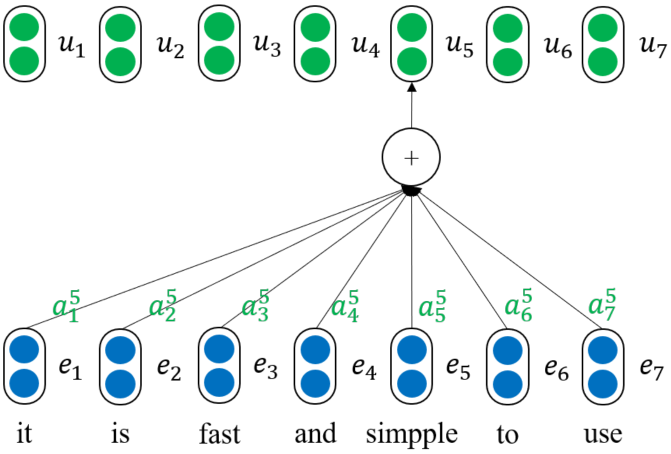


Figure 3.3: How to handle OOV words in the contextual attention mechanism

the other hand, Figure 3.3 shows how the vector representation of OOV word "simpple" is produced. Instead of using random vector in a traditional method, the contextual attention mechanism generates the vectors by summing the vectors of the context words such as "is", "fast", "to", and "use".

Intuitively, it is ineffective to consider function words or stop words in the contextual attention mechanism, since they has no explicit meaning and are independent with the polarity of the aspect term. As we mentioned above, the attention mechanism allow the model to focus on the suitable context words rather than non-function words (stop words). As a result, the value of α of stop words usually small.

Our aim is to create a contextual vector u_t that purely contains polarity information of opinion words. Stop words have no polarity in general. Although α of a stop word is usually small, it leads to decrease α of polarity words a little. It is preferable to estimate α of a stow word as zero, so that more information is conveyed from a polarity word to a target aspect. Therefore, we propose an additional technique to filter out these stop words and recalculate the contextual information as in Equation (3.22), (3.23) and (3.24).

$$\beta_i^t = \begin{cases} \alpha_i^t & \alpha_i^t \geq threshold \\ -100 & \alpha_i^t < threshold \end{cases} \tag{3.22}$$

$$\gamma_i^t = \frac{exp(\beta_i^t)}{\sum_j^T exp(\beta_j^t)} \tag{3.23}$$

$$u_t = \sum_{i=1}^{T} \gamma_i^t e_i \tag{3.24}$$

When α_i^t is small (less than *threshold*), i.e. the word at position i is a stopword, we set it to -100. It ensures that a revised parameter γ_i^t becomes 0. Then γ_i^t is used as a weight parameter in generation of the context vector u_t as shown in Equation (3.24). It means that the vectors of the stop words are not incorporated into u_t at all. Moreover, the weight parameter γ_i^t of content words will be larger after recalculating the weight parameters as shown in Equation (3.23). As a result, u_t contains more information from content words.

Figure 3.4 illustrates the overall architecture of BiLSTM+CRF with the contextual attention mechanism. u_t is used as an additional input of BiLSTM+CRF by concatenating it and the word embedding e_t. While the word embedding contains the meaning of the current word, the contextual vector contains the meaning of the context words especially opinion terms in the sentence. We call this model as CATT+BiLSTM+CRF where CATT means Contextual ATTention mechanism.

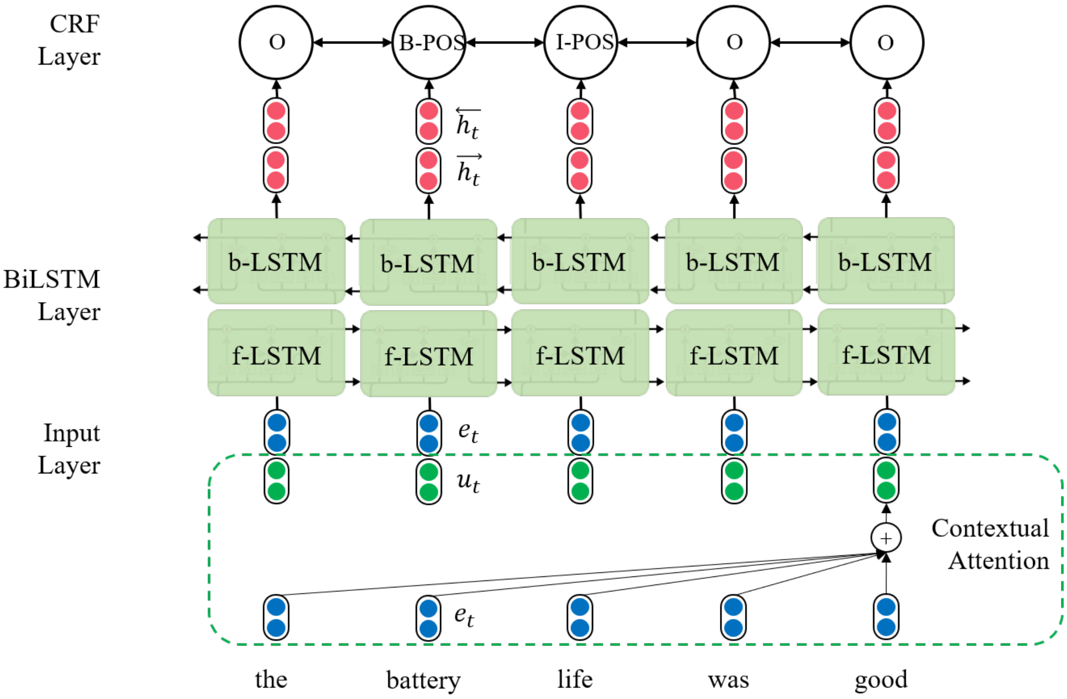


Figure 3.4: CATT+BiLSTM+CRF architecture

Chapter 4

Experiments

This chapter reports results of experiments to evaluate our proposed method. First, we first present datasets, evaluation metric, and experimental settings. Second, we show the effects of word embeddings on ATEPC models and describe the comparison between our proposed models and state-of-the-art models. Third, we visualize the attention score in CATT+BiLSTM+CRF to get a better understanding about the contextual attention mechanism. Finally, we perform error analysis to reveal advantages and disadvantages of the proposed model.

4.1 Data

We evaluate our proposed models BiLSTM+CRF and CATT+BiLSTM+CRF on four benchmark datasets from semantic evaluation competitions: SemEval 2014 [28], SemEval 2015 [27], and SemEval 2016 [26]. They are the collection of the reviews in the laptop and restaurant domains annotated with the aspect terms along with their polarity. The details of all datasets are illustrated in Table 4.1. Laptop14/Restaurant14, Restaurant15, and Restaurant16 are the dataset of SemEval 2014, 2015, and 2016, respectively.

Considering the difference of the domains of the datasets (i.e. laptop or restaurant), the word embeddings are pre-trained by Word2Vec [22] and GloVe [25] from different corpora. The Amazon dataset [1] [20] is used for training the embeddings on the laptop domain. Only reviews about electronic devices, which consists of 2.8 million sentences, are extracted and used for training. Meanwhile, the Yelp dataset [2] is used for training embeddings on the restaurant domain. This dataset contains 9.4 million sentences of users' opinions about restaurants and food.

[1] http://jmcauley.ucsd.edu/data/amazon

[2] https://www.yelp.com/dataset

Table 4.1: SemEval datasets

Datasets	Training no. sentences / no. aspect terms	Test no. sentences / no. aspect terms
Laptop14	3045 / 2358	800 / 654
Restaurant14	3041 / 3693	800 / 1134
Restaurant15	1315 / 1199	685 / 542
Restaurant16	2000 / 1743	676 / 622

4.2 Evaluation Metric

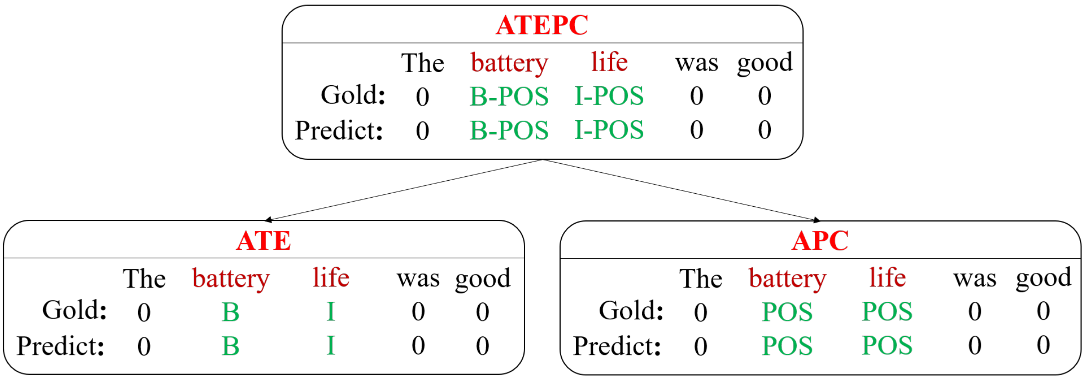
Although this paper proposes a joint model of two subtasks in ABSA, the performance of the models in the ATE and APC tasks are evaluated separately. After the results are produced by the model, we convert the output label back to the output of the ATE and APC tasks for comparison. Figure 4.1 illustrates this conversion. B_x and I_x, which are IOB labels of ATEPC, are converted to B and I to obtain results of ATE as shown in the left part of this figure. Furthermore, B_x and I_x are converted to the polarity label x to obtain results of APC as shown in the right part of Figure 4.1.

In general, a development data (validation data) is required to validate the training process of a deep learning model such as BiLSTM+CRF. In order to avoid overfitting, the number of epochs in iterative learning of parameters is determined so that the performance on a development data is maximized using early-stopping technique. We perform cross validation on training data for parameter optimization, where the data is divided into the training and development data. After training the model, a separated test data is used to evaluate it. There is another benefit of the cross validation. In general, the performance of the deep learning model is dependent on the initialization of parameters, since the initial parameters are usually set randomly. In the cross validation, initialization of parameters is performed several times so that the model is not overestimated or underestimated accidentally. In this experiment, we perform 10-fold cross validation on training data to obtain 10 optimized models. Then we apply these 10 models to the test data.

In the rest of this section, evaluation criteria for each ATE and APC task will be explained. When the results of the experiments are reported in Section 4.4 and 4.5, we show an average of 10 results in the evaluation criteria.

4.2.1 Evaluation criteria for ATE

The precision, recall, and F1-measure are used as evaluation criteria, since it is a kind of information extraction task. Precision, Recall, F1-measure are defined as in the following



ATEPC

	The	battery	life	was	good
Gold:	0	B-POS	I-POS	0	0
Predict:	0	B-POS	I-POS	0	0

ATE

	The	battery	life	was	good
Gold:	0	B	I	0	0
Predict:	0	B	I	0	0

APC

	The	battery	life	was	good
Gold:	0	POS	POS	0	0
Predict:	0	POS	POS	0	0

Figure 4.1: Converting output labels from ATEPC to ATE and APC

equations

$$P = \frac{|S \cap G|}{|S|} \tag{4.1}$$

$$R = \frac{|S \cap G|}{|G|} \tag{4.2}$$

$$F_1 = \frac{2 * P * R}{P + R} \tag{4.3}$$

, where S is a set of candidate aspect terms generated by a system for all test sentences, while G is a set of gold aspect terms. $S \cap G$ indicates a set of correct aspect terms which are recognised by a system.

4.2.2 Evaluation criteria for APC

We evaluate the performance of APC with respect to the accuracy, since it is a classification task. It is not meaningful to evaluate the polarity classification on incorrect aspect terms. Moreover, previous work usually removes conflict aspect terms which contain both positive and negative sentiment from test data. Therefore, we measure the accuracy of the APC task only for the aspect terms that are correctly extracted by the system and has no conflict of the polarity in the gold data. Equation (4.4) describes accuracy metric of the APC task.

$$accuracy = \frac{|A|}{|V|} \tag{4.4}$$

In this equation, $V \overset{def}{=} S \cap G$ indicates a set of correct aspect terms which are recognised by a system. A is a set of correct aspect terms whose polarity is correctly identified by a system. Obviously, A is a subset of V, i.e. $A \subseteq V$.

Comparison between two ATEPC models

Since the number of correct aspect terms that are recognised by two ATEPC models are usually different, it is somewhat unfair if we simply compare the accuracy of two ATEPC models. For example, if model #1 recognises 100 correct aspect terms and obtains 80% accuracy of APC, while model #2 recognises 120 correct aspect terms and obtains 79% accuracy of APC, we can not conclude that the model #1 is slightly better than the model #2 even the accuracy of the model #1 is better than that of the model #2. Therefore, we present a fair evaluation criteria of the APC task to compare two ATEPC models.

The *accuracy*$_1$ (the accuracy of the model #1) and *accuracy*$_2$ (the accuracy of the model #2) are defined as Equation (4.5) and (4.6).

$$accuracy_1 = \frac{|A_1|}{|V_1 \cap V_2|} \tag{4.5}$$

$$accuracy_2 = \frac{|A_2|}{|V_1 \cap V_2|} \tag{4.6}$$

V_1 and V_2 are sets of correct apsect terms recognized by the model #1 and #2. A_1 and A_2 are sets of correct aspect terms whose polarity is correctly identified by the model #1 and #2. Then *accuracy*$_1$ and *accuracy*$_2$ are compared to check which model is better. The basic idea for fair comparison is simple: we compare the accuracy with respect to only intersection of aspect terms correctly extracted by two models.

Comparison between two ATEPC models using cross validation

As described earlier, we perform 10-fold cross validation on training data to get 10 models and apply them to test data. It makes a fair comparison of two models difficult. It is not obvious how to evaluate which of two ATEPC models (such as BiLSTM+CRF and CATT+BiLSTM+CRF) is better when cross validation is performed. In this experiment, micro-accuracy is used as evaluation criteria of the APC task.

Let $M = \{m_1, m_2, ..., m_{10}\}$ is a set of 10 models obtained by a method #1 through 10-fold cross validation; $N = \{n_1, n_2, ..., n_{10}\}$ is a set of 10 models obtained by a method #2. The micro-accuracy$_1$ (the micro average of the accuracy of the method #1) and micro-accuracy$_2$ (that of the method #2) are defined as in Equation (4.7) and (4.8).

$$\text{micro-accuracy}_1 = \frac{\sum_{i=1}^{10} \sum_{j=1}^{10} |A_{m_i} \cap V_{m_i} \cap V_{n_j}|}{\sum_{i=1}^{10} \sum_{j=1}^{10} |V_{m_i} \cap V_{n_j}|} \tag{4.7}$$

$$\text{micro-accuracy}_2 = \frac{\sum_{i=1}^{10} \sum_{j=1}^{10} |A_{n_j} \cap V_{m_i} \cap V_{n_j}|}{\sum_{i=1}^{10} \sum_{j=1}^{10} |V_{m_i} \cap V_{n_j}|} \tag{4.8}$$

V_{m_i} and V_{n_j} are sets of correct aspect terms which are recognised by the model m_i and n_j while A_{m_i} and A_{n_j} are sets of correct aspect terms whose polarity is correctly identified by the model m_i and n_j, respectively. In the definition of micro-accuracy$_1$, for each pair of m_i and n_j, only aspect terms correctly extracted by two methods are taken account in the denominator. Within these aspect terms, ones whose polarity are correctly identified by a method #1 are taken account in the numerator. Note that all models m_i and n_j are applied to the same test data that is mutually exclusive with the training data. micro-accuracy$_2$ is defined in the same way.

4.3 Experimental Settings

We build our proposed models using Tensorflow [3], which is a powerful library for implementation of deep learning. We use Adam optimiser [14] instead of stochastic gradient descent (SGD) because it not only obtain better results but also take less time for training. To avoid overfitting, we use the dropout technique [33] after nonlinear layers, as well as the early-stopping technique. Moreover, we also allow the model to update the word embeddings during the training by setting "Embedding fine-tuning: True". Finally, we use zero padding to pad the input sequences with different lengths.

We optimize the hyperparameters of our model. Table 4.2 reveals a list of hyperparameters and their values to be investigated in the optimization procedure. The chosen (optimized) hyperparameters are shown in bold.

Table 4.2: Optimization of hyperparameters

Hyperparameter	Investigated values
Word embedding size (d_e)	100, **150**, 200
Word embedding fine-tuning	**True**, False
No. hidden units in BiLSTM+CRF (d_h)	100, **200**, 300
No. hidden units in CATT+BiLSTM+CRF (d_h)	100, 200, **300**
Optimiser	**Adam**, SGD
Learning rate	0.01, **0.001**, 0.0001
Zero padding	**True**
Attention size in CATT+BiLSTM+CRF (d_a)	100, 150, **200**
threshold in Equation (3.22) in CATT+BiLST+CRF	**0.01**, 0.02, 0.03
Dropout rate	0.4, **0.5**, 0.6, 0.7

To find optimized hyperparameters, we perform 10-fold cross validation on the training data. This procedure is similar to the one described in Section 4.2. However, instead of applying 10 models to the test data, we apply them to the development data and measure the sum of the F1-measure of ATE (Equation (4.3)) and the accuracy of APC (Equation (4.4)). An average for this index of 10 trials is used as an evaluation criterion to measure the performance of the model with each set of hyperparameters. Then the hyperparameters of the model that achieves the highest performance are chosen as the optimized ones.

[3] https://www.tensorflow.org

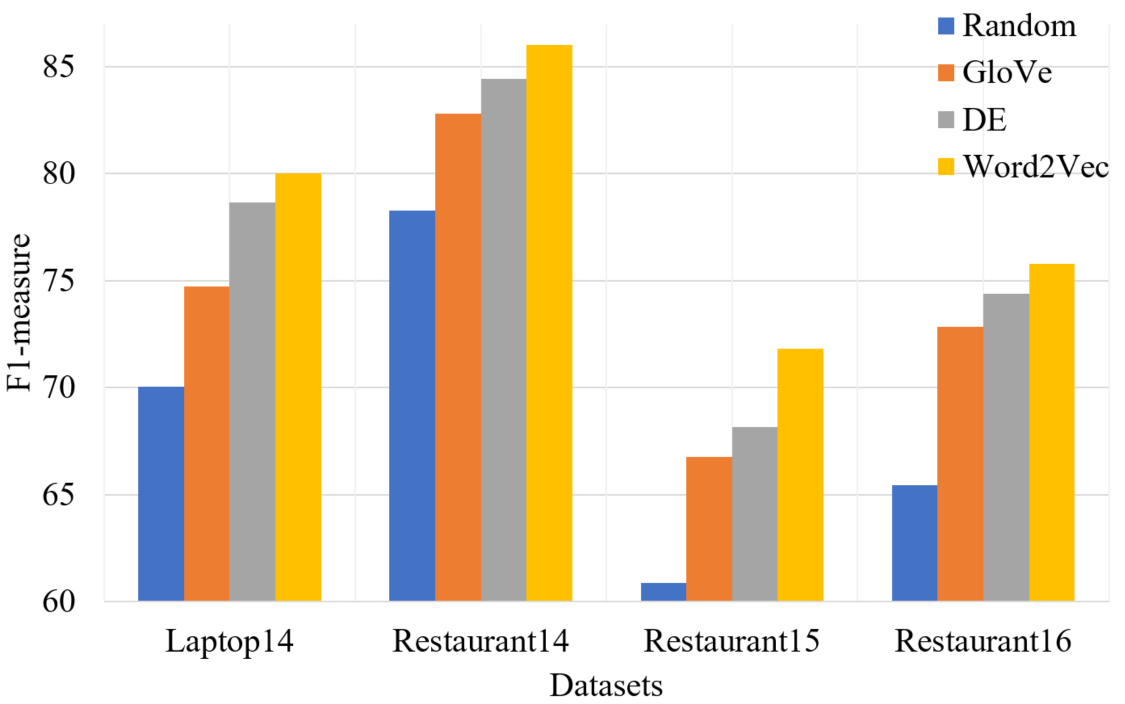


Figure 4.2: F-measure of the ATE task of BiLSTM+CRF with different word embeddings

4.4 Effectiveness of Word Embeddings

It is known that word embedding critically affects the performance of most deep learning models in natural language processing. It is due to the fact that word embedding captures a large number of precise syntactic and semantic properties of words so that the model can be easily optimized.. In this work, we conduct several experiments to evaluate effectiveness of different word embeddings on BiLSTM+CRF and CATT+BiLSTM+CRF. The following four types of word embedding are compared.

- **Random**: word vector where each dimension is set to a random value following a normal distribution of zero mean and 0.2 variance.
- **GloVe**: pre-trained word embedding on Yelp and Amazon datasets using GloVe [4].
- **Word2Vec**: pre-trained word embedding on Yelp and Amazon datasets using CBOW [5].
- **DE**: a combination of general word embedding [6] and domain-specific embedding [7].

[4] https://nlp.stanford.edu/projects/glove

[5] https://code.google.com/archive/p/word2vec

[6] The general word embedding is published at http://nlp.stanford.edu/data/glove.840B.300d.zip

[7] We use word embedding published at https://www.cs.uic.edu/~hxu . It is also trained from Amazon or Yelp dataset, which are used to train domain-specific word embeddings in Section 4.1.

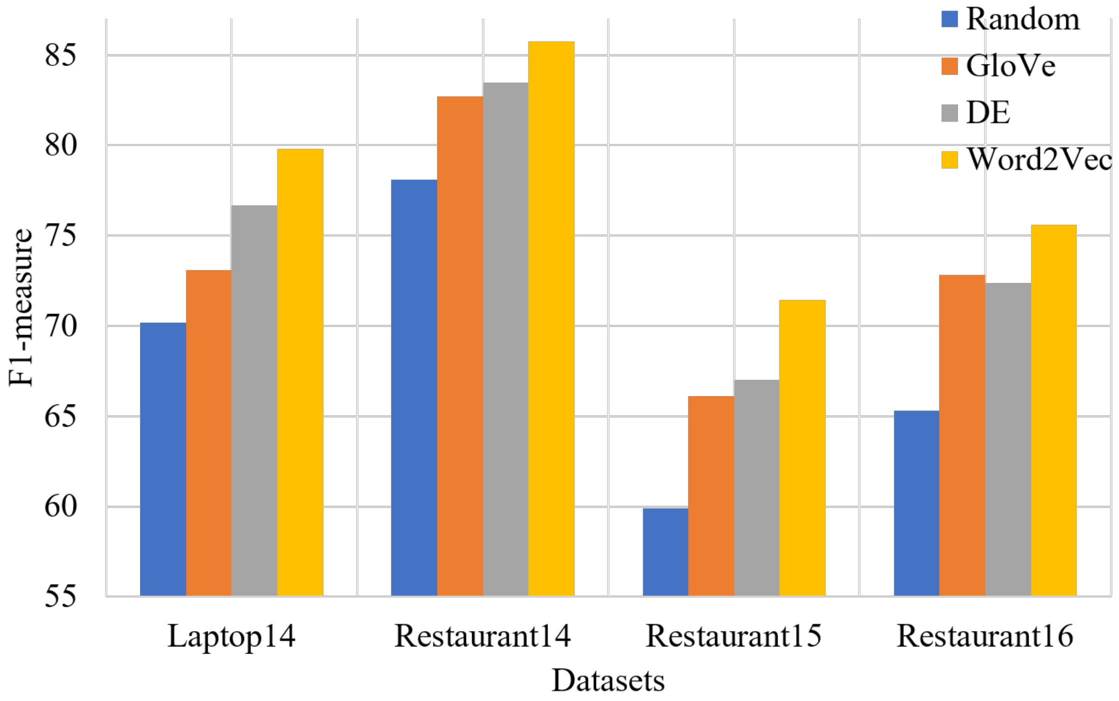


Figure 4.3: F-measure of the ATE task of CATT+BiLSTM+CRF with different word embeddings

We set dimension of word embedding to 150 for Random, GloVe, Word2Vec and allow the model to fine-tune these word embeddings in a training phase. On the other hand, DE uses the a concatenated vector of 300-dimensional general word embedding and 100 dimensional domain-specific word embedding. We do not fine-tune DE vector following Xu et al. [42]. Because they argued that fine-tuning would harm the word embeddings and decreased the performance of the model [8].

Figures 4.2 and 4.3 show F-measure of the ATE task of BiLSTM+CRF and CATT-BiLSTM+CRF using different word embeddings, respectively. It is clear that Word2Vec and DE achieve a better performance than GloVe and Random. It shows that context information plays an important role in learning word embedding which is a characteristic of CBOW (used to train Word2Vec) and fastText [9] (used to train DE) models.

Since Word2Vec outperformed the other methods, it is used as word embedding in the proposed model at the experiments in the next section.

[8] We tried to fine-tune the word embedding DE in the training phase, but the performance of the model became worse.

[9] https://github.com/facebookresearch/fastText

4.5 Evaluation of ATEPC models

In this section, we compare our proposed models with state-of-the-art models in the ATE and APC tasks.

4.5.1 Results of ATE task

This subsection reports results of the ATE task. We compare our models with the following the state-of-the-art baselines.

- **WDEmb** is a model proposed by Yin et al. [43] using a combination of word and dependency paths embeddings, along with some hand-crafted features as an input of CRF.
- **WE+POS+CNN** is a model proposed by Poria et al. [30] using word embedding and POS feature as inputs of a convolutional neural network.
- **DE+CNN** is a model proposed by Xu et al. [42] using general word embedding and domain-specific word embedding as inputs of a convolutional neural network. This model achieves the state-of-the-art result in the ATE task.
- **RNCRF+F** is a joint model proposed by Wang et al. [41] for aspect and opinion terms co-extraction (AOTE) task.
- **RNCRF-O+F** is a variation of RNCRF+F that performs only the ATE task without extraction of opinion terms.
- **CMLA** is a coupled multi-layer attention model proposed by Wang et al. [40]. This model is known as the state-of-the-art for the AOTE task.

Table 4.3 shows the results of these baselines as well as our proposed models BiLSTM+CRF, CATT+BiLSTM+CRF, BiLSTM+CRF-Pol, and CATT+BiLSTM+CRF-Pol. BiLSTM+CRF-Pol and CATT+BiLSTM+CRF-Pol are variations of the models proposed in Chapter 3 which perform only the ATE task. That is, the output tag set is $\{B, I, O\}$, which indicate the beginning, inside and outside of aspect terms, respectively. The results of the baselines are excerpted from the original papers.

The first group in Table 4.3 includes the models of the ATE task, while the second group includes the models of the AOTE task and our proposed models of the ATEPC task. The reason to distinguish the first and second groups is that dual propagation of aspect terms and opinion words can improve the performance of aspect term extraction [41]. In fact, RNCRF+F is better than RNCRF-O+F.

It is clear that our proposed models obtain good performance for the restaurant domain and promising performance for the laptop domain. Comparing with the models in the second group, BiLSTM+CRF outperforms RNCRF+F by 1.6 points for the Laptop14 dataset and CMLA by 0.71 and 1.11 point for the Restaurant14 and Restaurant15 datasets. Comparing with the models in the first group, BiLSTM+CRF outperforms WDEmb by 2.13 point for the Restaurant15 dataset and DE+CNN by 1.43 point

Table 4.3: F1 measure of ATE performance

Models	Laptop14	Restaurant14	Restaurant15	Restaurant16
WDEmb	75.16	84.97	69.73	
RNCRF-O+F	77.26	84.25		
RNCRF+F	78.42	84.93	67.47	
CMLA	77.80	85.29	**70.73**	
WE+POS+CNN*	81.06	**86.20**		
DE+CNN	**81.59**			**74.37**
BiLSTM+CRF	79.97	85.77	**71.84**	75.80
BiLSTM+CRF-Pol	80.44	86.10	71.79	76.39
CATT+BiLSTM+CRF	79.81	85.74	71.43	75.59
CATT+BiLSTM+CRF-Pol	**80.46**	**86.27**	70.90	**76.50**

* In original paper of WE+POS+CNN, they also show a better performance than this result using several linguistic patterns as a post-processing. However, in this case, we only show the result of their basic model to compare the ability of learning models.

for the Restaurant16 dataset. On the other hand, BiLSTM+CRF is comparable with WE+POS+CNN on the Restaurant14 dataset and worse than DE+CNN on the Laptop14 dataset. CATT+BiLSTM+CRF shows the similar performance as BiLSTM+CRF in comparison with the baseline methods.

Unexpectedly, our joint models BiLSTM+CRF and CATT+BiLSTM+CRF that solve the ATE and APC tasks simultaneously are slightly worse than BiLSTM+CRF-Pol and CATT+BiLSTM+CRF-Pol that solves ATE only. However, the difference between them is not statistically significant by the McNemar's test at 95% confidence level for both the restaurant and laptop datasets. This means that their performance is comparable. Furthermore, they outperform the state-of-the-art models on Restaurant15 and Restaurant16 datasets of the ATE task. Thus, our joint models are the promising method to extract aspect terms in ABSA.

Comparing the models with and without the contextual attention mechanism, BiLSTM+CRF performs better than CATT+BiLSTM+CRF by 0.16 point, 0.03 point, 0.4 point, and 0.21 point for Laptop14, Restaurant14, Restaurant15, and Restaurant 16 datasets respectively. These results are not statistically significant by the McNemar's test at 95% confidence level. It means that the contextual attention gives no contribution in boosting the performance of the ATE task.

Table 4.4 compares our proposed models with the state-of-the-art model WE+POS+CNN with respect to precision, recall, and F1-measure. Although our proposed models perform worse than WE+POS+CNN in terms of F1-measure and precision, they achieve better recall. In particular, CATT+BiLSTM+CRF outperforms WE+POS+CNN by 3.76 point and 2.24 point for the Laptop14 and Restaurant14 datasets respectively. Moreover, our proposed models obtain the balanced results between recall and precision, while WE+POS+CNN achieves high precision but low recall.

Table 4.4: Precision, recall, and F1-measure of ATE performance

Dataset	Models	Precision	Recall	F1-measure
Laptop14	WE+POS+CNN	**86.46**	76.31	**81.06**
Laptop14	BiLSTM+CRF	80.87	79.11	79.97
Laptop14	CATT+BiLSTM+CRF	79.58	**80.07**	79.81
Restaurant14	WE+POS+CNN	**87.42**	85.01	**86.20**
Restaurant14	BiLSTM+CRF	84.96	86.60	85.77
Restaurant14	CATT+BiLSTM+CRF	84.29	**87.25**	85.74
Restaurant15	BiLSTM+CRF	**71.56**	**72.17**	**71.84**
Restaurant15	CATT+BiLSTM+CRF	71.10	71.88	71.43
Restaurant16	BiLSTM+CRF	**75.40**	76.22	**75.89**
Restaurant16	CATT+BiLSTM+CRF	74.32	**76.96**	75.59

Table 4.5: Accuracy of APC performance

Models	Laptop14	Restaurant14	Restaurant15	Restaurant16
DCU	70.49	80.95		
Memnet	72.37	80.95		
RAM	74.49	80.23		
BiLSTM+CRF	69.26	80.99	79.53	87.12
CATT+BiLSTM+CRF	70.34	81.60	80.90	86.80

4.5.2 Results of APC task

In this subsection, we compare our proposed models with the state-of-the-art baselines in the APC task. Brief explanation of three baselines are shown below.

- **DCU** is a wining system in SemEval challenge 2014, task 4 (Aspect-based Sentiment Analysis), subtask 2 (Aspect Term Polarity). It is proposed by Wagner et al. [39] using hand-crafted features as an input of SVM.
- **Memnet** is a memory network for the APC task proposed by Tang et al. [37].
- **RAM** is a recurrent attention network on memory proposed by Chen et al. [3].

Table 4.5 shows results of the baselines as well as our proposed models. The results of the baselines are excerpted from the original papers. Note that the test data of our model and these previous models are not exactly the same. The classification of the polarity for the aspect terms correctly extracted by the system is evaluated for our models, while that of the given aspect terms is evaluated for the baselines.

It is found that BiLSTM+CRF and CATT+BiLSTM+CRF achieve comparable results of the state-of-the-art methods, although the comparison of the accuracy of APC in Table 4.5 is not completely fair. This indicates that the idea to perform ATE and APC simultaneously is promising for polarity classification of aspects.

Table 4.6: Micro-accuracy of BiLSTM+CRF and CATT+BiLSTM+CRF

Models	Laptop14	Restaurant14	Restaurant15	Restaurant16
BiLSTM+CRF	67.96	80.67	80.45	**87.55**
CATT+BiLSTM+CRF	**68.34**	**80.90**	**81.13**	87.30

Next, BiLSTM+CRF and CATT+BiLSTM+CRF are comapred by the micro-accuray defined in Subsection 4.2.2. Since the test samples are the exactly same in the calculation of micor-accuary, these two models can be fairly compared. Table 4.6 shows the micro-accuracy of 10 models of BiLSTM+CRF and 10 models of CATT+BiLSTM+CRF. It is found that CATT+BiLSTM+CRF outperforms BiLSTM+CRF by 0.38 point, 0.23 point, 0.68 point for Laptop14, Restaurant14, Restaurant15 datasets respectively. However, CATT+BiLSTM+CRF performs worse than BiLSTM+CRF by 0.22 point in the Restaurant15 dataset. Although these differences are small, they are statistically significant by the McNemar's test at 95% confidence level. It shows that the contextual attention mechanism plays a significant role in boosting APC performance of the model.

4.6 Attention Visualization

We show examples of attention actually learned by the model to clarify how the attention mechanism works. Figure 4.4 shows visualization of attention obtained for test sentences in the Laptop14 dataset. Each pixel in the figure stands for γ_i^t in Equation (3.23), which is the amount of information from word i to be incorporated into the current word t. That is, γ_i^t is the value at t-th row and i-th column. The darker the pixel is, the greater value of γ_i^t is. A black pixel indicates the highest value $\gamma_i^t = 1.0$, while a white pixel indicates the smallest value $\gamma_i^t = 0$.

We can see that the polarity information is directly added to each word in the sentence by using the contextual attention mechanism. For example, in the first sentence, although the opinion term "disappointment" is far from the aspect term "performance", the model is able to incorporate this information into "performance" position. In addition, the relationship between an aspect term and an opinion term is interpreted in the visualization. For instance, "windows 8" and "complaint" in the left-bottom graph as well as "baterry" and "longer" in the right-bottom graph has greater attention than other pixels in the same row. Furthermore, how to generate a word vector of out-of-vocabulary word is also graphically shown in the last example. "baterry" is a typographical error of "battery". The word vector of this OOV word is synthesized by a weighted sum of the word vectors in the context words, where the weights are defined by γ_i^t.

4.7 Error Analysis

We conducted an error analysis of BiLSTM+CRF and CATT+BiLSTM+CRF on the laptop dataset to investigate major causes of errors.

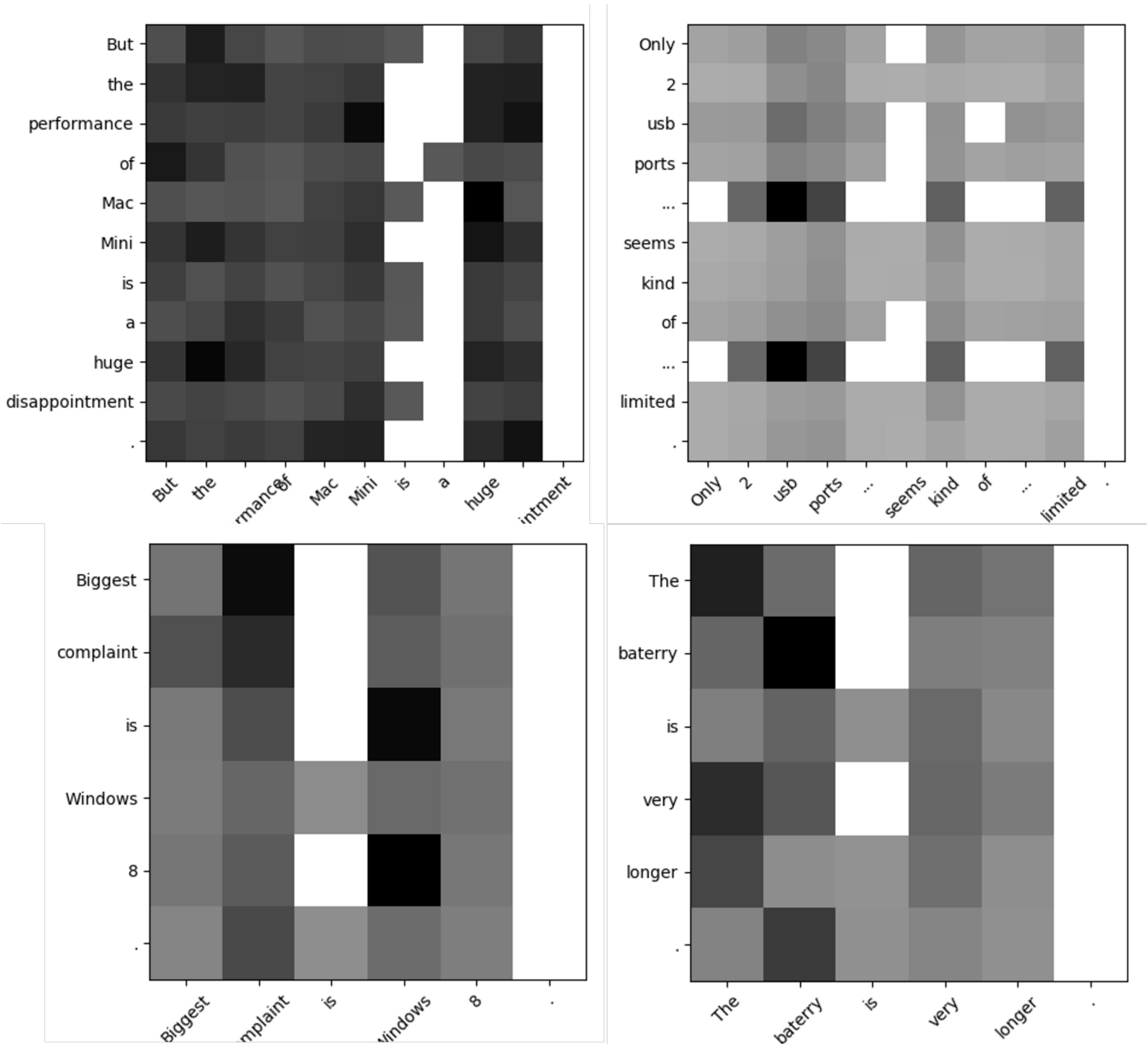


Figure 4.4: Attention visualization for some examples

As for ATE, the major problem was identification of an aspect term expressed by a phrase, such as in "It's fast at [loading the internet]."[10] However, this error can be acceptable because almost recognized words are headwords of the phrase of the aspect which play an important role in expressing the meaning of the aspect. The second most errors were false-negative errors on aspect terms of a verb or a named entity, such as in "It [looks] and [feels] solid, with a flawless [finish]" and "Not as fast as I would have expect for an [i5]." It is hard to extract these aspect terms since verbs and named entities are seldom appeared as aspect terms in the training data. Domain-specific knowledge is required to extract such aspect terms.

As for APC, first, many errors were caused in objective sentences where a user talked

[10] In the example sentences in this section, the gold aspect term is indicated by square brackets, while the predicted one is indicated by underline.

about a fact or description of a product without expressing an opinion. For example, in the sentence “Came with [iPhoto] and [garage band] already loaded,” the correct labels for these aspect terms were “neutral”, but the model classified them as “positive”. This is mainly caused by unbalanced datasets for the APC task where the number of “neutral” aspect terms was much fewer than that of “positive” and “negative”. Second, many errors were caused because the model could not handle negation appropriately. For instance, in the sentence “Not too expense and has enough [storage] for most users and many [ports],” these positive aspect terms were wrongly classified as negative by the model. Third, we found many errors in comparative sentences such as “this [operating system] beats [Windows] easily.”

Table 4.7 shows the example sentence with gold labels (aspect terms and its polarity), output labels by BiLSTM+CRF, and those by CATT+BiLSTM+CRF. It is clear that BiLSTM+CRF incorrectly recognises an aspect term and its polarity if the term is rare in the training dataset or misspelling (such as “complaint” in the first example, and “baterry” in the second example), or the opinion term is located in a long distance (such as “performance” and “disappointment” in the fourth example). Contextual attention mechanism is able to address these problems by incorporating the context information into aspect term position. However, if a sentence contains both positive and negative aspects in different phrases, the models will recognise the polarity incorrectly such as example 5 in the table.

Table 4.7: Examples of results obtained by BiLSTM+CRF and CATT+BiLSTM+CRF

1	Gold	Biggest complaint is [Windows 8]$_{NEG}$.
	BiLSTM+CRF	Biggest complaint is $\underline{\text{Windows 8}}_{POS}$.
	CATT+BiLSTM+CRF	Biggest complaint is $\underline{\text{Windows 8}}_{NEG}$.
2	Gold	The [baterry]$_{POS}$ is very longer .
	BiLSTM+CRF	The $\underline{\text{baterry}}_{NEG}$ is very longer .
	CATT+BiLSTM+CRF	The $\underline{\text{baterry}}_{POS}$ is very longer .
3	Gold	Only 2 [usb ports]$_{NEG}$ ... seems kind of ... limited .
	BiLSTM+CRF	Only 2 $\underline{\text{usb ports}}_{POS}$ ... seems kind of ... limited .
	CATT+BiLSTM+CRF	Only 2 $\underline{\text{usb ports}}_{NEG}$ ... seems kind of ... limited .
4	Gold	But the [performance]$_{NEG}$ of Mac Mini is a huge disappointment .
	BiLSTM+CRF	But the $\underline{\text{performance}}_{POS}$ of Mac Mini is a huge disappointment .
	CATT+BiLSTM+CRF	But the $\underline{\text{performance of Mac Mini}}_{NEG}$ is a huge disappointment .
5	Gold	air has higher [resolution]$_{POS}$ but the [fonts]$_{NEG}$ are small .
	BiLSTM+CRF	air has higher $\underline{\text{resolution}}_{POS}$ but the $\underline{\text{fonts}}_{POS}$ are small .
	CATT+BiLSTM+CRF	air has higher $\underline{\text{resolution}}_{NEG}$ but the $\underline{\text{fonts}}_{NEG}$ are small .

Chapter 5

Conclusions

5.1 Summary

This thesis introduced a new task named ATEPC, which aimed to extract aspect terms and their polarity simultaneously. This task not only overcame the problem of resources and time consuming in a sequential model (ATE and APC are performed as separate modules) but also partly prevented the model from causing chain errors that are conveyed from ATE to APC in a sequential model.

We first conducted the experiments on the ATEPC task with a model named BiLSTM+CRF that is a combination of bidirectional LSTM and conditional random field (CRF). Although BiLSTM+CRF is a common sequential tagging model, it expressed a powerful performance in the ATEPC task through our experiments. The empirical results on four datasets showed that our proposed model (BiLSTM+CRF) achieved several state-of-the-art performance in the ATE task and it produced promising results in the APC task.

We also proposed the contextual attention mechanism to improve the APC performance. The major advantage of this technique was that it directly considered polarity information of opinion terms that were far from the target aspect term. In addition, a minor advantage of the attention mechanism was handling out-of-vocabulary words. Instead of using random representation for out-of-vocabulary words, the contextual attention mechanism created a better representation for them by synthesizing a context vector which was a weighted sum of the word vectors of the context words. The experiments showed that the model with the contextual attention mechanism (CATT+BiLSTM+CRF) was comparable with BiLSTM+CRF in the ATE task. Moreover, it achieved a significant improvement in the APC task compared with BiLSTM+CRF. Furthermore, the visualization of attention for some example sentences revealed that the contextual attention mechanism was a promising method to incorporate polarity information from a long distance.

5.2 Future Work

In future, a new loss function that is more appropriate to both ATE and APC tasks should be investigated. In addition, handling subjective sentences and negation that cause most of the errors in the current model should be explored to improve the performance of the APC task. Moreover, using convolutional neural network for the ATEPC task is also a promising direction because it performs well in the ATE task in the previous work. Finally, because the contextual attention visualization showed the relation between aspect and opinion terms, we believe that it is totally suitable for the task of aspect and opinion terms co-extraction (AOTE). An AOTE model with the contextual attention mechanism should be implemented and empirically evaluated.

Bibliography

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[2] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *Trans. Neur. Netw.*, 5(2):157–166, March 1994.

[3] Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 452–461. Association for Computational Linguistics, 2017.

[4] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.

[5] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, November 2011.

[6] Franck Dernoncourt, Ji Young Lee, and Peter Szolovits. NeuroNER: an easy-to-use program for named-entity recognition based on neural networks. *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, 2017.

[7] Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 49–54, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[8] Alex Graves and Jrgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *NEURAL NETWORKS*, pages 5–6, 2005.

[9] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 388–397, Vancouver, Canada, July 2017. Association for Computational Linguistics.

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

[11] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA, 2004. ACM.

[12] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991, 2015.

[13] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 151–160, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[15] John F. Kolen and Stefan C. Kremer. *Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies*. IEEE, 2001.

[16] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June 2016. Association for Computational Linguistics.

[17] Chenghua Lin and Yulan He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 375–384, New York, NY, USA, 2009. ACM.

[18] Bing Liu. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2012.

[19] Bing Liu and Lei Zhang. *A Survey of Opinion Mining and Sentiment Analysis*, pages 415–463. Springer US, Boston, MA, 2012.

[20] Julian McAuley and Alex Yang. Addressing complex and subjective product-related queries with customer reviews. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 625–635, Republic and Canton of Geneva, Switzerland, 2016. International World Wide Web Conferences Steering Committee.

[21] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. Topic sentiment mixture: Modeling facets and opinions in weblogs. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 171–180, New York, NY, USA, 2007. ACM.

[22] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[23] Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. Recurrent models of visual attention. *CoRR*, abs/1406.6247, 2014.

[24] Thien Hai Nguyen and Kiyoaki Shirai. PhraseRNN: Phrase recursive neural network for aspect-based sentiment analysis. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 2509–2514, 2015.

[25] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543. ACL, 2014.

[26] 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 *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California, June 2016. Association for Computational Linguistics.

[27] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495, Denver, Colorado, June 2015. Association for Computational Linguistics.

[28] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University.

[29] Ana-Maria Popescu and Oren Etzioni. Extracting product features and opinions from reviews. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 339–346, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[30] Soujanya Poria, Erik Cambria, and Alexander Gelbukh. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, 108:42 – 49, 2016. New Avenues in Knowledge Bases for Natural Language Processing.

[31] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *CoRR*, abs/1509.00685, 2015.

[32] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Stroudsburg, PA, October 2013. Association for Computational Linguistics.

[33] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January 2014.

[34] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3104–3112, Cambridge, MA, USA, 2014. MIT Press.

[35] Charles Sutton and Andrew McCallum. An introduction to conditional random fields. *Found. Trends Mach. Learn.*, 4(4):267–373, April 2012.

[36] Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. Effective lstms for target-dependent sentiment classification. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 3298–3307, 2016.

[37] Duyu Tang, Bing Qin, and Ting Liu. Aspect level sentiment classification with deep memory network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 214–224, 2016.

[38] Ivan Titov and Ryan McDonald. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 111–120, New York, NY, USA, 2008. ACM.

[39] Joachim Wagner, Piyush Arora, Santiago Cortes, Utsab Barman, Dasha Bogdanova, Jennifer Foster, and Lamia Tounsi. Dcu: Aspect-based polarity classification for semeval task 4. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 223–229, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University.

[40] Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeie, and Xiaokui Xiao. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3316–3322, 2017.

[41] Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. Recursive neural conditional random fields for aspect-based sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 616–626, 2016.

[42] Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. Double embeddings and cnn-based sequence labeling for aspect extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 592–598. Association for Computational Linguistics, 2018.

[43] Yichun Yin, Furu Wei, Li Dong, Kaimeng Xu, Ming Zhang, and Ming Zhou. Unsupervised word and dependency path embeddings for aspect term extraction. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 2979–2985. AAAI Press, 2016.

[44] Wayne Xin Zhao, Jing Jiang, Hongfei Yan, and Xiaoming Li. Jointly modeling aspects and opinions with a maxent-lda hybrid. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 56–65, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.