A Study on Deep Learning Models for Sentiment Analysis in Hospitality Media

TRAN Xuan Thang

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

March, 2019

Master's Thesis

A Study on Deep Learning Models for Sentiment Analysis in Hospitality Media

1710121 TRAN Xuan Thang

Supervisor: Professor HUYNH, Nam-Van
Main Examiner: HUYNH, Nam-Van
Examiners: Yukio Hayashi
DAM Hieu Chi
Yasuo Sasaki

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
[Knowledge Science]

February, 2019

Abstract

With the high-speed development of the Internet in this information age, more and more people over the world can easily access the Internet and use websites to share their opinions, feelings, experiences or complains about products or services on social media (e.g., forum discussion, blogs and social networks, travel websites). For both businesses and customers, these contents (such as reviews, tweets, blogs,...) are the worthiest sources for not only enhancing their services, launching new products, making business campaigns but also making their decisions when choosing a specific service or product. However, the explosion of number websites on the Internet had led to the difficulty of manually monitoring the massive volume of opinionated reviews. Additionally, the analyses of experts towards those opinionated text of the services and products are diverse and heterogeneous and may cause biases. To overcome these limitations, researchers have explored a new research direction in natural language processing area named sentiment analysis or opinion mining that is able to analyze opinionated text systematically and automatically.

According to The World Travel and Tourism Council, Travel and Tourism is the fast-growing industry and become the key to every country's economy. Tourism becomes one of the world's largest economic sector by supporting over 300 million global jobs and contributing more than 10 percent of world GDP. In the tourism sector, hospitality industry has occupied the most significant percentage. Specifically, hospitality industry covers accommodations, food and beverages, travels, transportations, and other fields of tourism sector. Hospitality industry contributes multibillion-dollar annually by providing specific services, products, and experiences for customers based on the leisure time and disposable income of customers. Simultaneously with the explosion of social media, travelers and customers who used those services and products regularly post their comments, opinions, and experiences on websites. Reviews of travelers regarding different characteristics of a service or product bring the advantages for the managers to improve their business, and for the potential customers before choosing a service or product. In finding a way to improve the practical experience of both buy-side and sell-side in the hospitality market, we apply document-level sentiment analysis for hospitality data collected from a user-generated content site named TripAdvisor. Typically, from big data including both

quantitative data and qualitative data of customer's reviews, we apply some deep learning models to identify the customers' sentiment opinions expressed in the text reviews.

The contributions of this thesis are first conducting a comprehensive investigation into sentiment analysis and recent related work. Secondly, we investigate the architecture of some deep learning models, including Recurrent Neural Network, Long Short-Term Memory, Gated Recurrent Unit, Bidirectional Long Short-Term Memory, and Convolutional Neural Network, which frequently applied on current research of sentiment analysis using machine learning approach, and they had archived the state-of-the-art results in these tasks. Additionally, we collected a new hospitality media dataset including more than 75,000 reviews of 410 hotels in Ho Chi Minh City, Vietnam up to February 2017 from a well-known, world's largest travel site named TripAdvisor. Finally, we carried out experiments using those above deep learning models with the new dataset to evaluate the performance of them for the complicated and complex task of sentiment analysis task.

As the results showed, although we just employ some general deep learning models with slight configurations in the architectures, they can perform well with the complicated task. Deep learning models averagely classified 89.95 percent correctly of the sentiment opinions on the reviews given from the hospitality media dataset. They completely overperformed some baseline models, including Support Vector Machine and Naiv̈e Bayes, which were also famous in sentiment analysis field. This study adds more contributions to finding the emerging opinions of customers towards the hotel reviews by applying deep learning algorithms which ultimately benefits for further studies in this area.

Keywords: sentiment analysis - machine learning - deep leaning models - hospitality media - hotel industry.

Acknowledgments

With boundless love and appreciation, I would like to extend my heartfelt gratitude and appreciation to the people who have supported me, not only finishing this Master dissertation, but throughout my Master study.

First and foremost, I would like to express the deepest appreciation to my supervisor, Professor HUYNH Van Nam from Japan Advanced Institute of Science and Technology (JAIST), for his patience, motivation, and immense knowledge. Without his guidance and persistent help, this thesis would not have been possible.

I would like to acknowledge my examination committee: Prof. Yukio Hayashi, Prof. DAM Hieu Chi, Prof. Yasuo Sasaki, and my second supervisor Prof. Takaya Yuizono, who gave me a lot of suggestions to improve my thesis. In addition, I would like devote my deepest gratitude to Mr. NGUYEN Ba Hung, a doctoral student at Huynh-lab, JAIST, for his guidance, paper revision, and encouragement through my Master study.

I gratefully acknowledge the funding source that made my graduate study abroad possible. I received a full-ride scholarship of Project 599 established by Ministry of Education and Training, Vietnam. Moreover, I would like to express my gratitude to Japan Advanced Institute of Science and Technology for letting me fulfill my dream of being a student here.

I would also like to thank my colleagues at Informatics Department, Faculty of Natural Sciences and Technology, Tay Nguyen University, Vietnam for sharing the busy teaching work and giving me a opportunity to pursue Master degree at JAIST.

Lastly, I would like to thank my friends, Huynh-lab members, especially Yu Yang, and my family for all their friendship, understanding, wisdom, patience, enthusiasm, and encouragement and for pushing me farther than I thought I could go.

Thank you,

TRAN, Xuan Thang

JAIST, February 2019

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

We first introduce the background of our research in Section 1.1. The research motivations and research goals are explained in Section 1.2. Then we introduce the contributions of this study in Section 1.3. Finally, we illustrate the thesis structure in the Section 1.4.

1.1 Background

Sentiment analysis (SA), or opinion mining, is a research area of study that analyzes opinionated text in natural language processing area. Sentiment analysis aims to determine people's opinions, evaluations, attitudes, appraisals, and emotions towards entities such as services, products, events, individuals, organizations, and their attributes [1]. This task is useful to not only individuals in making decisions but also organizations in improving their services or launching new products. For example, potential customers can determine the advantages and disadvantages of services before choosing by discovering the reviews from other purchased customers. Companies always want to capture their customer's opinions to enhance their services, facilities, and marketing campaigns of their products before launching into the market.

According to The World Travel and Tourism Council[1], Travel and Tourism is the fast-growing industry, and is a key to every country's economy. With the number of tourists growing year by year, tourism becomes one of the world's largest economic sector, supporting 313 million worldwide jobs and generating 10.4% of world GDP annually.

[1]https://www.wttc.org/

Travel and Tourism has been expected to support more than 0.4 billion jobs over the world, which equates to 1 in 9 of all worldwide jobs, and it will contribute around 25% of global net job creation over the next decade. In the tourism sector, hospitality industry takes the central place. The hospitality industry is a main business of the service industry that provides accommodations, foods and beverages, travel and tourism, transportations, and others within the tourism industry. Nowadays, hospitality is a multibillion-dollar industry which focused on the satisfaction of customers by providing specific experiences for customers.

With the proliferation of user-generated content (UGC) on the Internet, many consumers use websites to share their feelings, experiences, or complains about services, products, or trip destinations [2]. User-generated content is the online platform where customers reflect their thoughts and raise their ratings for products and services [3]. Online opinionated feedbacks (such as reviews, tweets, blogs...) in UGC take an essential place for potential customers to make their decisions [4]. Especially, customer's feedback is an integral part of the continuous improvement process implemented in the hotel industry, and yet a comprehensive characterization of the customer experience is difficult to achieve. Nowadays, consumers are participating and spending more time on social media to make friends, co-create, share information, experiences, and opinions [5]. Their purchase as a decision-making process is being influenced by those factors through social media networks. Hence, managers are focusing on online communication platforms to reach the online consumers and to take advantages of their feedbacks in the social networks.

Nowadays, with the massive development of the Internet and social media, tourists can easily express their feelings or their opinions about the accommodations, restaurants, attractive tourist attractions,... and services during their trips on some UGC sites (e.g. SNS sites such as Facebook [2], Twitter[3], or travel website such as Booking[4], TripAdvisor[5]). They are the worthiest sources for not only the managers to enhance their services, facilities, and marketing campaigns but also for potential customers to consider services and products before making the decisions. For example, according to TripAdvisor's Investor

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

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

[4]https://booking.com

[5]https://tripadvisor.com

Relations[6], more than 702 million reviews and opinions had been uploaded to TripAdvisor up to November 07 2018, and TripAdvisor-branded sites are home to the world's largest travel community of 490 million averaged monthly unique visitors. However, there is a challenge and difficulty in monitoring and analyzing such kinds of information manually due to a massive amount of opinionated text reviews on social media. Moreover, the analyses of human towards these opinionated text reviews of services and products are diverse and heterogeneous and are easily led to significant biases, e.g. people often enjoy the reviews concerning to their preferences [6]. Therefore, these limitations can be avoided by using a automated sentiment analysis system. Instead of using hand-crafting methods, researchers become more interested in analyzing opinionated text automatically by using some machine learning techniques. Recently, machine learning has occupied a central place and expanded on many research fields and intelligent systems. Particularly, deep learning becomes the hottest trend in machine learning thank to its computational efficiency on matrix operations and the stronger computing power of the computer. These characteristics will enable deep learning models to shine in analyzing big-unstructured data, especially in sentiment analysis to understand the sentiment opinions and the targets of a text document. These deep learning models can overcome the limitations of subjective opinion biases and the shortages of opinionated text that can be caused by human.

1.2 Research Motivation

This research aims to conduct a comprehensive investigation into machine learning models for sentiment analysis on hospitality review data. Particularly, we mainly focus on deep learning models and their applications in sentiment analysis. Based on this investigation, this research will determine the architectures of some previously developed deep learning models and then points out the advantages and disadvantages of them. Our research goal is to help hotel managers gain more accurate, useful information and insights from their customers and services. By the motivation from the above discussion, we first explore the task definition of this research, then decide the appropriate algorithms based on deep learning. Furthermore, we will conduct experiments on the new hospitality dataset

[6]http://ir.tripadvisor.com/

crawled from TripAdvisor to evaluate the performance of different deep learning models for sentiment analysis in hotel reviews.

1.3 Research Contributions

The contribution of this thesis is as follows:

- We conduct a comprehensive survey about sentiment analysis, techniques to solve the sentiment analysis task, and current research on sentiment analysis. Especially, we classify the differences, advantages, and drawbacks of some deep learning models applying for document-level sentiment analysis, including Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), Bidirectional Long Short-Term Memory (BiLSTM), and Convolutional Neural Network (CNN). We also introduce the word embedding technique to map a word (text) to a vector (number), which is then fed into deep learning models at the input phrase.
- We collect a new hospitality media dataset on TripAdvisor, which covers all English reviews for 410 hotels in Ho Chi Minh City, Vietnam up to February 2017, and do some descriptive analyses.
- We conduct experiments for those deep learning models on the new dataset to show that deep learning models archive better performances in sentiment analysis for hospitality media compared with some previous models (baseline models).

1.4 Thesis Outline

The rest of this thesis is organized as follows:

- In chapter 2, we conduct a comprehensive survey about sentiment analysis, deep learning, and recent work on the sentiment analysis task, especially in hotel industry.
- In chapter 3, we first introduce a framework for sentiment analysis using deep learning models. This is a general framework which mostly applied for supervised learning tasks using deep learning algorithms. In addition, we describe some deep learning

models which are well-known for sentiment analysis task on document level, including Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), Bidirectional Long Short-Term Memory (BiLSTM), and Convolutional Neural Network (CNN).

- In chapter 4, we introduce a new hospitality media dataset which collected from TripAdvisor and then do some descriptive statistics. Furthermore, we config the experimental settings for the deep learning models, and present the experimental results of different deep learning models for SA task on the new dataset.

- In chapter 5, we summarize the contribution of this study, and determine some limitations, future work for the sentiment analysis task in hospitality media.

Chapter 2

Related Work

2.1 Sentiment Analysis

According to Bing Liu [1], Sentiment analysis (SA), or *opinion mining*, is the field of study that analyzes human' opinions, sentiments, feelings, emotions... towards entities such as products, services, organizations, individuals... Sentiment analysis mainly focuses on opinions which express or imply positive or negative sentiments. Since 2000, sentiment analysis has emerged to become the most attractive and active research field in natural language processing (NLP) area.

Sentiment analysis can be performed on three different levels: document-level, sentence-level, and aspect-level. At document-level, sentiment analysis focus on detecting whether an opinionated text expressed positive or negative opinion by considering the whole document as an essential information unit (only discussing unique topic). Sentence-level sentiment analysis pays more attention to classify sentiment opinion expressed on sentences of a long document using two steps. First, it identifies whether the sentence is subjective or objective. While a subjective sentence usually expresses personal sentiment or opinion, an objective sentence gives some factual information about an object, such as product or service [1]. The next step is to classify whether the subjective sentence expressed positive or negative opinion. The study of Wilson et al. [7] pointed out that sentiment opinions can be expressed in both subjective and objective sentences in nature. However, because sentence is likely just a short document, the difference between them is not fundamental [1]. An example of sentence-level sentiment analysis is shown as follows:

Example 1: Posted by Tiano on Sept-10-2010

(1)I stayed in Nikkon Hotel last trip. (2)The room was clean and the wifi was strong. (3)However, it was a bit noisy at night. (4)So, I changed the hotel on the next day.

Given Example 1, a sentiment analysis system will classify whether a sentence expressed positive, negative, or no sentiment opinion. From above example, this system will classify sentence (1) as no opinion (or neutral), sentence (2) as positive opinion, and sentence (3) as negative opinion respectively. Especially, although sentence (4) "So, I changed the hotel on the next day." is an objective sentence (because it gave the true fact), it implicitly expressed negative feeling about the hotel. Thus, it is more suitable to determine every sentence in a long document as opinionated sentence or not, rather than subjective or objective sentence.

On the other hand, Aspect-level sentiment analysis, as known as Aspect-based sentiment analysis - ABSA, provides necessary details about all aspects of a entity mentioned in text documents, which is crucial in many applications. ABSA aims to classify the sentiment concerning specific aspects of a entity. For aspect-level sentiment analysis, we use a quintuple $(e_i, a_{ij}, o_{ijkl}, h_k, t_l)$ [1] to represent an opinion, where:

- e_i denotes an entity's name.
- a_{ij} denotes an aspect term belong to entity e_i.
- h_k is a opinion holder who expressed the opinion.
- t_l indicates the specific time when h_k expressed the opinion.
- o_{ijkl} is the opinion towards aspect a_{ij} of entity e_i given by h_k person at t_l time.

The opinion o_{ijkl} can be classified as (*pos, neg, neu*) levels, or indicated by different numerical/intensive measurement, such as star ratings, or numerical numbers. When the opinion holder h_k did not express the opinion o_{ijkl} towards any specific aspect a_{ij}, we denote aspect a_{ij} of entity e_i as ***GENERAL***. We show an example of opinion quintuples as follows:

Example 2: Posted by bigAbc on Dec-20-2017

(1)I bought a Nokia cellphone and my brother bought a Samsung cellphone last month. (2)We kept in touch when we went to Tokyo. (3)The speech of my Nokia was perfect, but the battery life is short. (4)My brother was quite satisfied with his Samy, and with its sound . (5)I also want a cellphone like my brother. (6)So am going to return it tomorrow.

Four opinion quintuples will be generated from the above example:

– (Nokia, sound_quality, positive, BigAbc, Dec-20-2010)

– (Nokia, battery_life, negative, BigAbc, Dec-20-2010)

– (Samsung, sound_quality, positive, BigAbc's_brother, Dec-20-2010)

– (Samsung, GENERAL, positive, BigAbc's_brother, Dec-20-2010)

Each quintuple describes the entity, aspect, sentiment opinion, opinion holder, and posted time respectively regarding to the definition of $(e_i, a_{ij}, o_{ijkl}, h_k, t_l)$. According to Bing Liu [1], six following tasks must be performed to extract the opinion quintuples from document D:

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

  In this task, a system will extract all entities mentioned in opinionated document D and group synonymous entities into entity clusters. Each cluster points out an unique entity e_i. For instance, the system must extract three entities "Nokia", "Samsung", "Samy" from Example 2, and then categorizes them into two groups "Nokia", and "Samsung", in which "Samsung" and "Samy" belong to the unique cluster "Samsung".

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

  This task is to point out all aspects of the above entities, and then group synonymous aspects into same terms. Each aspect term a_{ij} represents a particular aspect of entity e_i. For example, three aspects "speech", "battery life", and "sound" must be extracted from Example 2, and then be classified into two unique aspect clusters "battery_life" and "sound_quality", where "speech" and "sound" are merged into "sound_quality".

- **Task 3: Opinion holder extraction**

  In this task, a system extracts the name of people who gave the opinions in document D. From Example 2, "BigAbc" and "BigAbc's_brother" are extracted as opinion holders.

- **Task 4: Time extraction**

  This task aims to extract the time t_l when the opinion holder h_k gave the opinions in document D. From Example 2, "Dec-20-2010" is extracted as time.

- **Task 5: Aspect sentiment classification**

  This task is to determine the sentiment opinion o_{ijkl} for each aspect a_{ij} of entity e_i. The sentiment an be classified as positive, negative, neutral or numeric sentiment ranking. From Example 2, the polarity of "battery_life" of "Nokia" expressed by "BigAbc" on "Dec-20-2010" is negative, while the rest aspects are classified as positive.

- **Task 6: Opinion quintuple generation**

  This is the final task for ABSA that produces all opinion quintuples expressed in the document D based on the results given from about tasks.

There are many techniques to solve the sentiment analysis tasks (as known as sentiment classification tasks) . According to Medhat et al. [8], there are two main approaches for sentiment classification, including machine learning and lexicon-based approach as in Figure 2.1. The approach using *Lexicon-based* technique is built on a collection of precompiled sentiment terms, and can be classified as two approaches. The first dictionary-based approach uses statistical methods to identify the sentiment opinion consisting on the document. The other approach called corpus-based approach focuses on semantic techniques to archive the same goal. On the other hand, the *Machine learning Approach (ML)* applied some famous machine learning models with linguistic features. The machine learning approach for sentiment classification can be classified as supervised and unsupervised learning. While supervised learning uses a massive amount training dataset with labels to train models and automatically detect the sentiment polarity, unsupervised leaning is employed when the labeled dataset is not available.

In this study, we mainly focus on applying deep learning models for sentiment analysis

task on document level. A new hospitality media dataset collected from a UGC platform named TripAdvisor will be used to train and test these models to deepen our knowledge of the performance of deep learning models for the complicated and complex task of understanding the customer's opinions in hotel reviews.

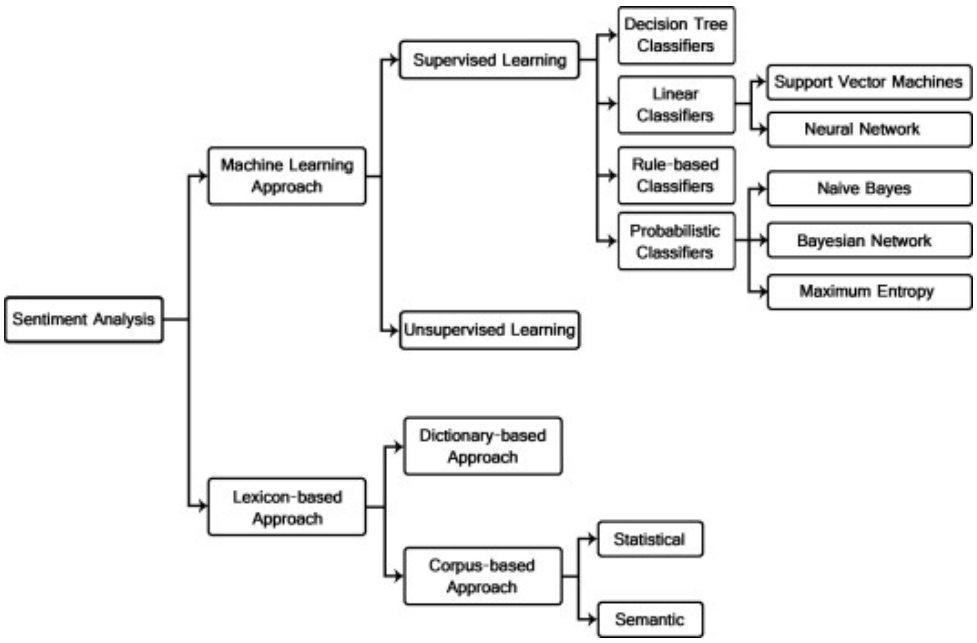

Figure 2.1: Sentiment classification techniques.

2.2 Deep Learning

Deep learning is a part family of supervised learning that aims to represent the features at different level of abstractions. By combining multiple hidden layers (usually more than three), deep learning can capture the abstract representations of every feature of large dataset. The fundamental idea of deep learning is to use the backpropagation algorithm to change the model hyperparameters to compute the abstract representation using the information from previous hidden layers [9]. Recently, deep learning has occupied the central place in almost research fields of computer science, and it had archived the state-of-the-art in image recognition [10, 11], speak recognition [12, 13], natural language processing,... In NLP, deep learning models have archived high performances, as in sentiment analysis tasks [14,15], question answering system [16], automated machine translation systems [17, 18]. It is clear that deep learning models, even without using any hand-crafting

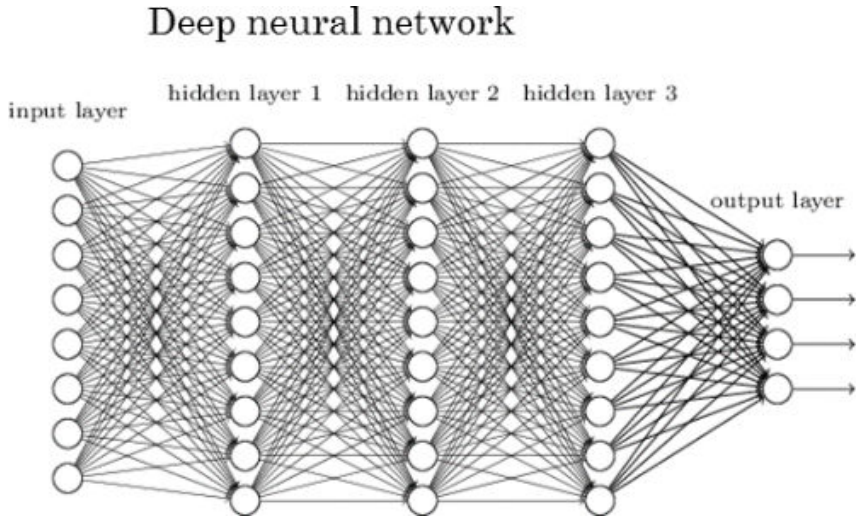

Figure 2.2: An example of deep learning network with 3 hidden layers.

knowledge, had outperformed lexicon-based methods and feature-based methods [19, 20]. The primary advantages of deep learning are its computational efficiency on matrix operations and its ability to learn the latent structured representation of big-unstructured data in vector space. Figure 2.2 illustrates an example of a fully-connected deep learning network with three hidden layers [21], where each node from a layer is fully connected with every node from previous layer. In this thesis, we will investigate some deep learning models for sentiment analysis task in Section 3.3.

Chapter 3

Methodology

In this chapter, we first introduce a framework using deep learning models for sentiment analysis. Then we explain the technique to convert word into vector named word embedding, which is well-known for sentiment analysis tasks. Section 3.3 shows the architecture of some deep learning models for sentiment analysis, including RNN, LSTM, GRU, BiLSTM, and CNN.

3.1 Framework

We use a general framework which widely apply for document-level sentiment analysis task using deep learning models. Figure 3.1 illustrates the whole process, from collecting dataset to training, test the models, and classifying the sentiment opinions of hotel reviews.

First, we collect a new hospitality media dataset by using an open framework named Scrapy[1] for extracting text reviews from TripAdvisor[2]. Then we apply some techniques in NLP to preprocess the data to get the cleaned dataset, such as noise data cleaning, language detection (only keep English reviews), punctuation, stopwords removal, tokenization, ... Because the input of almost machine learning models, especially deep learning models is numerical data, so we must convert the sequential text reviews into numerical matrices. The process called vectorization will be applied in this phrase to con-

[1]https://scrapy.org/

[2]https://tripadvisor.com

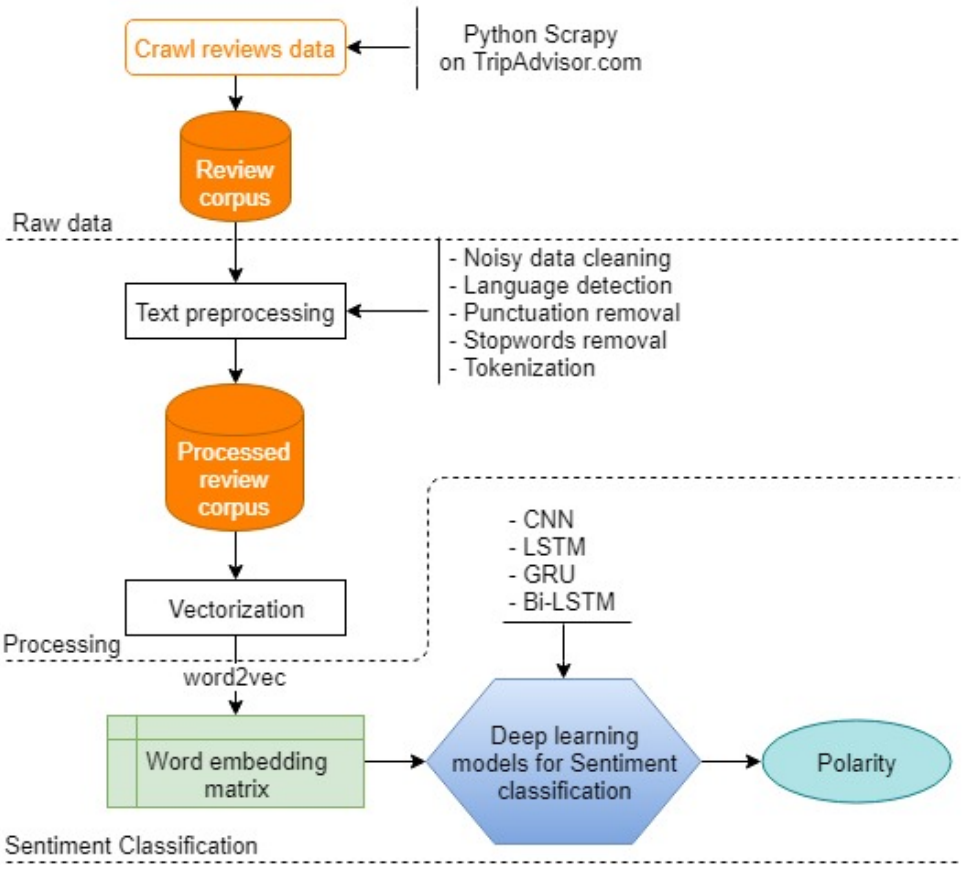

Figure 3.1: Framework for sentiment analysis task on document-level.

vert each word into its vector by using a pre-trained model named Google Word2Vec[3] to get the word embedding matrices. Google Word2Vec is a pre-trained model trained using Google News dataset (100 billion words) which widely uses in many research in natural language processing area. The model consists three million unique words and phrases, where each of them is represented using a unique 300-dimensional vector.

The word embedding matrices given from previous stage will be fed into deep learning models to get the output, which is the polarity of each sentence (classified as positive, negative). In this research, we apply some well-known deep learning models, including CNN, LSTM, GRU, BiLSTM with some tuning techniques to get the final results, in order to give the comparison. Figure 3.2 illustrates the basic mechanism using supervised machine learning models for sentiment analysis task on text reviews [22] (document-level or sentence-level sentiment analysis). Applying into our research, we first train

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

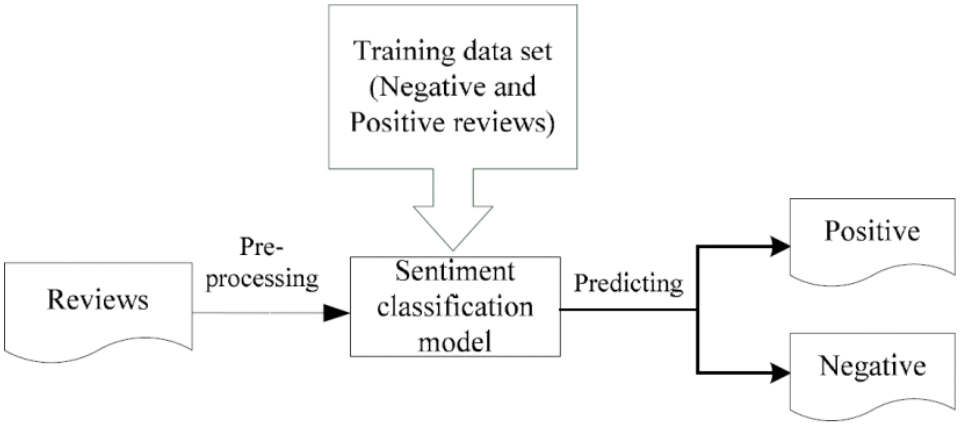

Figure 3.2: The flow chart of supervised learning for sentiment classification task.

sentiment classification models (deep learning algorithms) using a training dataset. Then, the processed reviews will be fed into these models to determine the polarity (pos, neg sentiment). The results of these experiments will be shown in Chapter 4.

3.2 Word Embedding

Word embedding is a high-quality distributed vector representation that captures a large number of precise syntactic and semantic properties of words. The main ideal of word embedding is to group similar words together [23] to help machine learning models archiving better performance in NLP tasks. In vector space, the synonymous words will be assigned in the adjacent locations. These learned vectors explicitly encode many linguistic regularities and patterns. For instance: vector("Hanoi") - vector("Vietnam") + vector("Japan") is closer to vector("Tokyo") than other words in vector space. Figure 3.3 illustrates some examples of relations between words in vector space. The right most figure shows the allocation of countries and their capitals in the same vector space. Figure 3.3 also illustrates the relationship of male-female and verb tense (past form and progressive form):

vector("man") - vector("queen") + vector("man") $\approx$ vector("woman")
vector("swimming") - vector("swam") + vector("walking") $\approx$ vector("walked")

There are two popular methods to train word embedding in the natural language

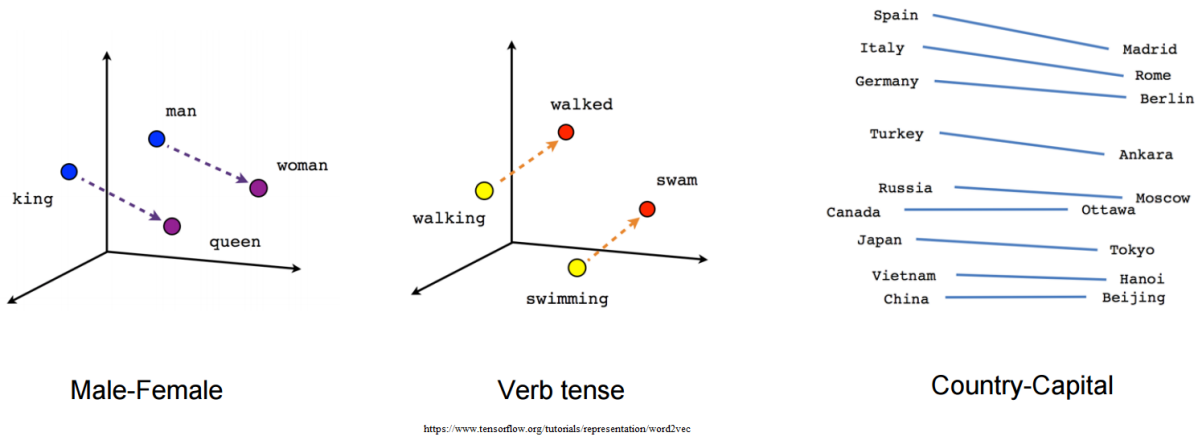

Figure 3.3: A visualization about the relation between words in vector space.

processing area, including Word2Vec and GloVe[4]. Both Word2Vec and GloVe can capture the semantic and syntactic of words. However, Word2Vec preserves semantic analogies for basic arithmetic on the word vectors, e.g. vec("Hanoi") - vec("Vietnam") + Vec("Japan") is closer to vec("Tokyo"), while GloVe preserves semantic analogies for global word-word co-occurrence statistics in a corpus. Recent research on NLP tasks has confirmed that both Word2Vec and GloVe are effective in almost studies. In this thesis, we investigate Word2Vec method to vectorize the text document to numerical matrices for deep learning model's inputs.

3.3 Deep learning models for Sentiment Analysis

3.3.1 Recurrent Neural Network

Recurrent neural network (RNN) is a part family of neural networks that performs well with the input data is interdependent such as a speech, transactional data or language. The original neural network assumes that the relations of inputs and outputs of the model are not exist. However, in some tasks using sequential data especially in natural language processing, the computation at a stage must be related to the previous stages. For example, if a model want to forecast the next word of a sentence, it is better to know which words expressed previously. The fundamental idea of RNN is to perform exactly

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

task for every component in a sequential set (e.g. sentence, speech, time-serial data), where the output of a state is depended on the previous computations. The original CNN model [9] can be illustrated as in Figure 3.4 (left diagram). After unrolling (right diagram), the RNN model can be understood as the multiple copies of a same network at different sequential time steps, which uses the output from previous time step as the input information. For example, if a sentence has three words, RNN model can be unfolded into a 3-layer neural network as in Figure 3.5 .

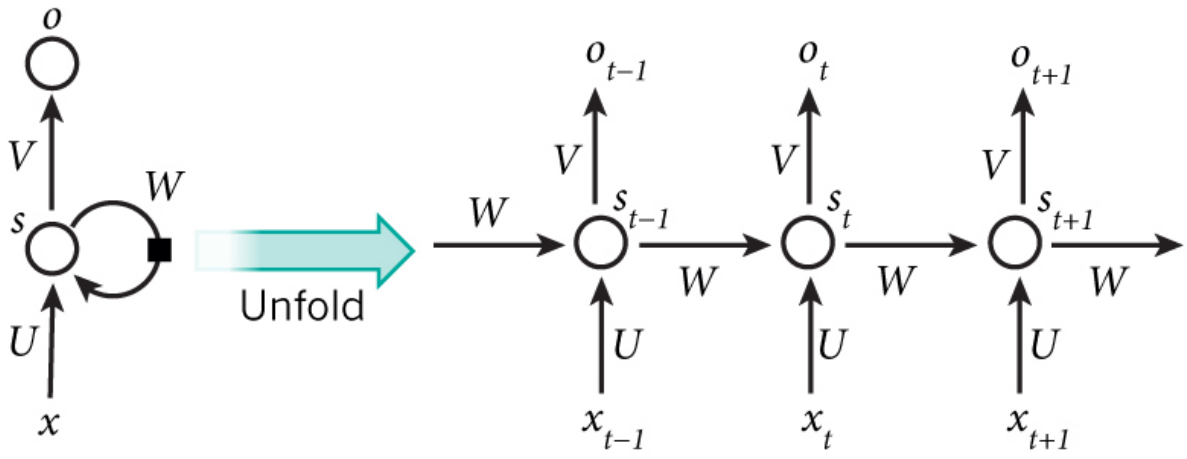

Figure 3.4: A original RNN model and the unfolding of the RNN model.

Given an input sentence $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$ where $\mathbf{x}_t$ is input vector at time step t and N is the length of the sentence, RNN performs recursively at each time step as in Equation 3.1.

$$\mathbf{s}_t = f\left(\mathbf{U} \cdot \mathbf{x}_t + \mathbf{W}_{rnn} \cdot \mathbf{s}_{t-1} + \mathbf{b}_s\right) \tag{3.1}$$

$$\mathbf{o}_t = softmax\left(\mathbf{V} \cdot \mathbf{s}_t + \mathbf{b}_o\right) \tag{3.2}$$

Parameters in this equation are explained as follows:

- $\mathbf{s}_t$ indicates hidden state of time step t and $\mathbf{h}_0$ is initialized as zero vector. $\mathbf{s}_t$ can be understood as "memory" cell of RNN and is calculated using the previous hidden layer and the input at the current time step t. Activation function f can be $tanh$ or $ReLU$ [24].

- $\mathbf{o}_t$ is the output of model at step t.

- $\mathbf{W}_{rnn}$, $\mathbf{V}$, $\mathbf{U}$, and $\mathbf{b}$ are the model hyperparameters which will be changed and optimized during the training process.

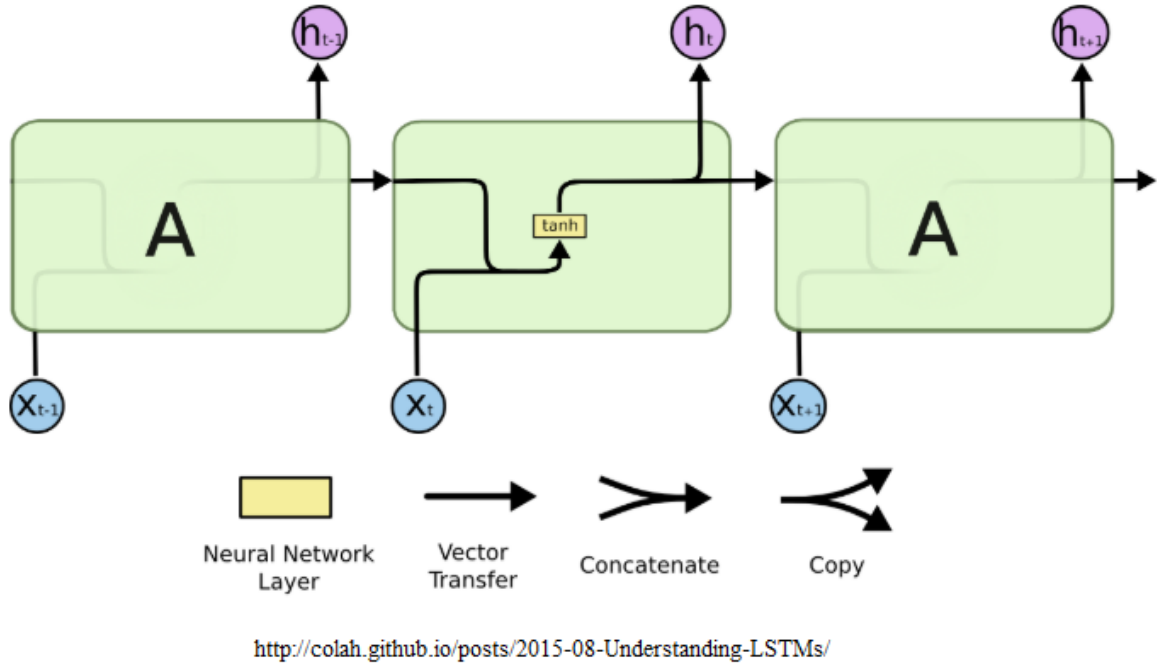

Figure 3.5: Visualizing the standard RNN architecture with three time steps.

At each time step, RNN network combines the current input information $\mathbf{x}_t$ and the past information from previous time step $\mathbf{s}_{t-1}$ to create a new vector representation for current state $\mathbf{h}_t$. For that reason, RNN model can perform well on sequential dataset such as text document. For the sentiment analysis tasks, the final output $\mathbf{o}_{final}$ of the last hidden state containing the whole information of the sequence will be classified as positive or negative class using $softmax$ activation function.

In theory, RNN model can capture long-term dependent information using the recursive architecture. However, the practical experiments showed that the RNN model had been suffered from the vanishing gradient problem. During the training process, the hyperparameters of model are not updated in back propagation phrase due to the tiny change in gradient value. It led to the early stop of iterative learning of RNN's hyperparameters [25]. As the result, the tasks using long-sequential dataset (long documents, reviews) could not archive the optimized performances with RNN model. To overcome this situation, a new version of RNN named Long Short-Term Memory has been proposed.

3.3.2 Long Short-Term Memory

A variation of RNN had been proposed by Hochreiter and Schmidhuber [26] to overcome the limitations of RNN. Currently, LSTM model has been widely employed in different tasks in many research fields of computer science. With the significant improvement in the architecture, LSTM had archived the state-of-the-art many current studies, such as automated machine translation [18], speech recognition [27], named entity recognition [28, 29], and sentiment analysis [30]. The main difference compared with RNN is the way LSTM model remembering the past information for a long time. Particularly, LSTM uses a "memory" cell to remember information that is far from the current state.

LSTM also has a recursive architecture as RNN, but it uses a different way to process information at each time step. While RNN uses one neural network layer to process the information, LSTM uses four layers, or 4 gates to automatically process the information. The processed information is then accumulated in "memory" stage after every time state. For a given input sentence $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$, where $\mathbf{x}_t \in \mathbb{R}^{d_i}$ is d_i-dimensional input vector at time state t and N is the length of sentence, the equations as follows exploit the model's structure at each recursive step of LSTM model.

$$\mathbf{f}_t = \sigma\left(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f\right) \tag{3.3}$$

$$\mathbf{i}_t = \sigma\left(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i\right) \tag{3.4}$$

$$\mathbf{o}_t = \sigma\left(\mathbf{W}_o \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o\right) \tag{3.5}$$

$$\widetilde{\mathbf{C}}_t = f\left(\mathbf{W}_C \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_C\right) \tag{3.6}$$

$$\mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \widetilde{\mathbf{C}}_t \tag{3.7}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh\left(\mathbf{C}_t\right) \tag{3.8}$$

The parameters in these equations are explained as follows:

- $\mathbf{f}, \mathbf{i}, \mathbf{o}, \mathbf{C} \in \mathbb{R}^{d_h}$ indicate a forget gate, an input gate, an output gate, and a cell state ("memory" state) respectively. d_h is number of hidden units in each LSTM cell.
- $\mathbf{h}_t \in \mathbb{R}^{d_h}$ stands for hidden state of model, which is also known as the output state of a LSTM unit.

- $\mathbf{W} \in \mathbb{R}^{d_h \times (d_h + d_i)}$ and $\mathbf{b} \in \mathbb{R}^{d_h}$ stand for the the model hyperparameters, which will learned and optimized by an optimization method during the training process, such as Adam optimization.

- f is an activation function, which usually is the hyperbolic tangent tanh. σ is also an activation function named sigmoid function. Both of tanh and σ are non-linear functions. The difference between them is that tanh function scales the output to the range $[-1, 1]$, while σ activation results in output that is between $[0, 1]$.

- $\odot$ represents an element-wise product, and the square brackets stand for the concatenation operator.

Figure 3.6 illustrates a graphical example of LSTM model with three time steps. The core component of LSTM is the "memory" cell state C which stores information through all time steps. However, "memory" cell does not have the ability to remove or add information into it directly. It must use three gates including f, i, o to decide how much information can be stored or deleted to/from the "memory". The forget gate determines the amount of information in "memory" should be removed. The input gate decides the amount of new information to be saved in "memory". The output gate considers how much the old information from the"memory" and the new information from the input at time state t could sent to the hidden layer of LSTM unit. For sentiment analysis task, the $\mathbf{h}_{final}$ at the final recursive step of LSTM will be classified by $softmax$ activation function to get the final result.

Many current research are widely employing LSTM model and its variations, and has archived remarkable performances in many NLP systems. The principle of LSTM model showed that it can easily handle long-term dependency and the vanishing gradient problem compared with RNN model. Moreover, LSTM network can archive better performances in many tasks without using any hand-crafting knowledge. In this work, we employ LSTM with some slight configurations as one of deep learning models due to the advantages that LSTM can learn the abstract representation of words in documents.

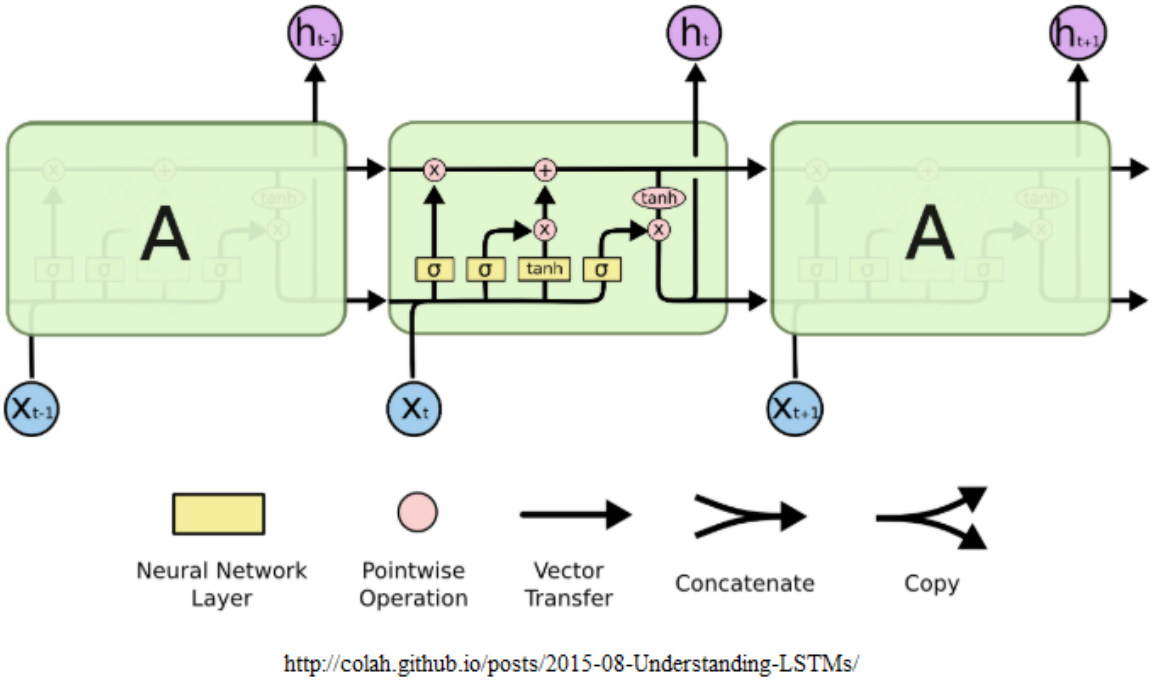

Figure 3.6: Visualizing the LSTM architecture with three time steps.

3.3.3 Gated Recurrent Units

In this thesis, we also investigate a dramatic variation of LSTM introduced by Cho et al. [31] named Gated Recurrent Unit (GRU). In this model, the forget gate f and the input gate i had been merged into a new "update" gate, and the "memory" and the hidden state are also combined into a new "reset" gate. By merging and reducing the number of gates, GRU model is now simpler than standard LSTM model, and has been steadily popular. Figure 3.7 shows the architecture of a GRU unit.

The following equations illustrate each recursive step in GRU for a given input sentence $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$ where $\mathbf{x}_t \in \mathbb{R}^{d_i}$ is d_i-dimensional input vector at time step t and N is the length of the sentence.

$$\mathbf{z}_t = \sigma\left(\mathbf{W}_z \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_z\right) \tag{3.9}$$

$$\mathbf{r}_t = \sigma\left(\mathbf{W}_r \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_r\right) \tag{3.10}$$

$$\widetilde{\mathbf{h}}_t = \tanh\left(\mathbf{W} \cdot [\mathbf{r}_t \odot \mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_h\right) \tag{3.11}$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \widetilde{\mathbf{h}}_t \tag{3.12}$$

These parameters in these equations are explained as follows:

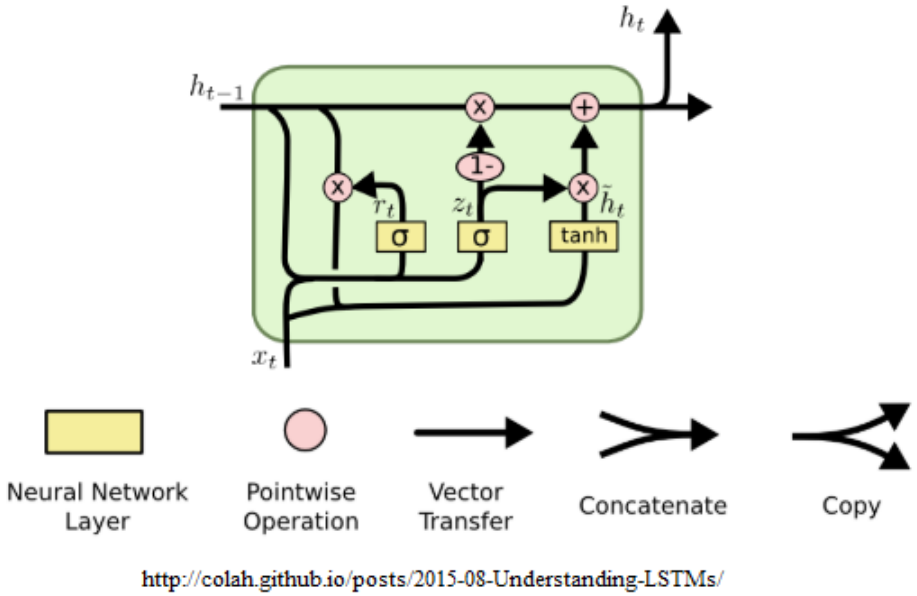

Figure 3.7: Visualizing the architecture of GRU.

- $\mathbf{r}, \mathbf{z} \in \mathbb{R}^{d_h}$ stand for a reset gate and a update gate respectively. d_h is the number of hidden units in GRU model.
- $\mathbf{h}_t \in \mathbb{R}^{d_h}$ stands for hidden state of model, which is also known as the output state of a GRU unit.
- $\mathbf{W} \in \mathbb{R}^{d_h \times (d_h + d_i)}$ and $\mathbf{b} \in \mathbb{R}^{d_h}$ are the model hyperparameters which are changed and optimized by an optimization function, such as Adam optimization.
- tanh is the activation function and σ is the sigmoid function. Both of them are activation functions as explained in LSTM model.
- $\odot$ represents an element-wise product, and the square brackets stand for the concatenation operator.

In sentiment analysis task, the $\mathbf{h}_{final}$ at the final recursive step of GRU will be classified by $softmax$ activation function to the final result.

3.3.4 Bidirectional Long Short-Term Memory

A famous variant of LSTM called Bidirectional Long Short-Term Memory (BiLSTM) was first published by Graves et al. [32]. Given a sentence $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ with n

words, where $\mathbf{x}_t \in \mathbb{R}^{d_i}$ is the d_i-dimensional vector of a word at time step t, a forward LSTM model captures the representation $\overrightarrow{\mathbf{h}}_t$ from left to right (the past) of a word in sentence. To capture the idea that the representation from the future of a word (from the right to the left in the sentence) also brings the benefit to train the model, BiLSTM uses the second LSTM (backward) in the reverse direction (compared to the forward LSTM) to compute a representation $\overleftarrow{\mathbf{h}}_t$ given the input sentence $(\mathbf{x}_n, \mathbf{x}_{n-1}, \ldots, \mathbf{x}_2, \mathbf{x}_1)$. Recently, BiLSTM archived significant results in NLP, such as named entity recognition [33], sentiment analysis [34], online handwriting recognition [35]. Figure 3.8 illustrates the architecture of BiLSTM.

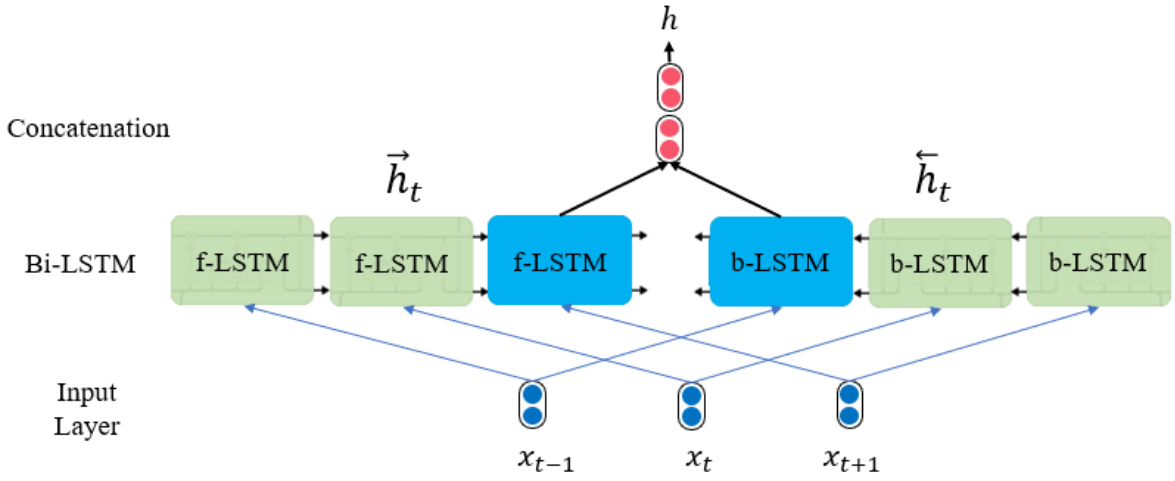

Figure 3.8: Visualizing the BiLSTM architecture with three time steps.

Each step of BiLSTM can be calculated as follows:

$$\overrightarrow{\mathbf{f}}_t = \sigma\left(\overrightarrow{\mathbf{W}}_f \cdot \left[\overrightarrow{\mathbf{h}}_{t-1}, \mathbf{x}_t\right] + \overrightarrow{\mathbf{b}}_f\right) \tag{3.13}$$

$$\overrightarrow{\mathbf{i}}_t = \sigma\left(\overrightarrow{\mathbf{W}}_i \cdot \left[\overrightarrow{\mathbf{h}}_{t-1}, \mathbf{x}_t\right] + \overrightarrow{\mathbf{b}}_i\right) \tag{3.14}$$

$$\overrightarrow{\mathbf{o}}_t = \sigma\left(\overrightarrow{\mathbf{W}}_o \cdot \left[\overrightarrow{\mathbf{h}}_{t-1}, \mathbf{x}_t\right] + \overrightarrow{\mathbf{b}}_o\right) \tag{3.15}$$

$$\overrightarrow{\tilde{\mathbf{C}}}_t = \tanh\left(\overrightarrow{\mathbf{W}}_C \cdot \left[\overrightarrow{\mathbf{h}}_{t-1}, \mathbf{x}_t\right] + \overrightarrow{\mathbf{b}}_C\right) \tag{3.16}$$

$$\overrightarrow{\mathbf{C}}_t = \overrightarrow{\mathbf{f}}_t \odot \overrightarrow{\mathbf{C}}_{t-1} + \overrightarrow{\mathbf{i}}_t \odot \overrightarrow{\tilde{\mathbf{C}}}_t \tag{3.17}$$

$$\overrightarrow{\mathbf{h}}_t = \overrightarrow{\mathbf{o}}_t \odot \tanh\left(\overrightarrow{\mathbf{C}}_t\right) \tag{3.18}$$

$$\overleftarrow{\mathbf{f}}_t = \sigma\left(\overleftarrow{\mathbf{W}}_f \cdot \left[\overleftarrow{\mathbf{h}}_{t-1}, \mathbf{x}_t\right] + \overleftarrow{\mathbf{b}}_f\right) \tag{3.19}$$

$$\overleftarrow{\mathbf{i}}_t = \sigma\left(\overleftarrow{\mathbf{W}}_i \cdot \left[\overleftarrow{\mathbf{h}}_{t-1}, \mathbf{x}_t\right] + \overleftarrow{\mathbf{b}}_i\right) \tag{3.20}$$

$$\overleftarrow{\mathbf{o}}_t = \sigma\left(\overleftarrow{\mathbf{W}}_o \cdot \left[\overleftarrow{\mathbf{h}}_{t-1}, \mathbf{x}_t\right] + \overleftarrow{\mathbf{b}}_o\right) \tag{3.21}$$

$$\overleftarrow{\widetilde{\mathbf{C}}}_t = \tanh\left(\overleftarrow{\mathbf{W}}_C \cdot \left[\overleftarrow{\mathbf{h}}_{t-1}, \mathbf{x}_t\right] + \overleftarrow{\mathbf{b}}_C\right) \tag{3.22}$$

$$\overleftarrow{\mathbf{C}}_t = \overleftarrow{\mathbf{f}}_t \odot \overleftarrow{\mathbf{C}}_{t-1} + \overleftarrow{\mathbf{i}}_t \odot \overleftarrow{\widetilde{\mathbf{C}}}_t \tag{3.23}$$

$$\overleftarrow{\mathbf{h}}_t = \overleftarrow{\mathbf{o}}_t \odot \tanh\left(\overleftarrow{\mathbf{C}}_t\right) \tag{3.24}$$

Parameters in these equations are explained as follows:

- Both $\overrightarrow{\mathbf{h}}_t, \overleftarrow{\mathbf{h}}_t \in \mathbb{R}^{d_h}$ are the hidden states of model. They are also known as the output state of a BiLSTM unit, where $\overrightarrow{\mathbf{h}}_t$ is the output of forward LSTM model and $\overleftarrow{\mathbf{h}}_t$ is for backward LSTM model respectively. d_h is the number of hidden units in LSTM.

- $\overrightarrow{\mathbf{W}} \in \mathbb{R}^{d_h \times (d_h + d_i)}$ and $\overrightarrow{\mathbf{b}} \in \mathbb{R}^{d_h}$ are the model hyperparameters of the forward LSTM, while $\overleftarrow{\mathbf{W}} \in \mathbb{R}^{d_h \times (d_h + d_i)}$ and $\overleftarrow{\mathbf{b}} \in \mathbb{R}^{d_h}$ are the model hyperparameters of the backward LSTM.

- $\odot$ represents an element-wise product, and the square brackets stand for the concatenation operator.

The final result from forward LSTM $\overrightarrow{\mathbf{h}}_{final}$ and backward LSTM $\overleftarrow{\mathbf{h}}_{final}$ are concatenated into a new long vector, and then passed through a $softmax$ activation function to compute the final classification result.

3.3.5 Convolutional Neural Network

Convolutional neural networks (CNN) is a part of deep artificial neural networks, which most applied to local features [36]. Formerly invented for computer vision, convolutional neural network algorithm has achieved the state-of-the-art performances on some NLP tasks related to sentiment classification [14, 37, 38]. Another impressive results can be observed at semantic parsing tasks [39], information retrieval [40], sentence modeling [41], and other tasks in NLP [42]. By transforming the tokens consisting each sentence into

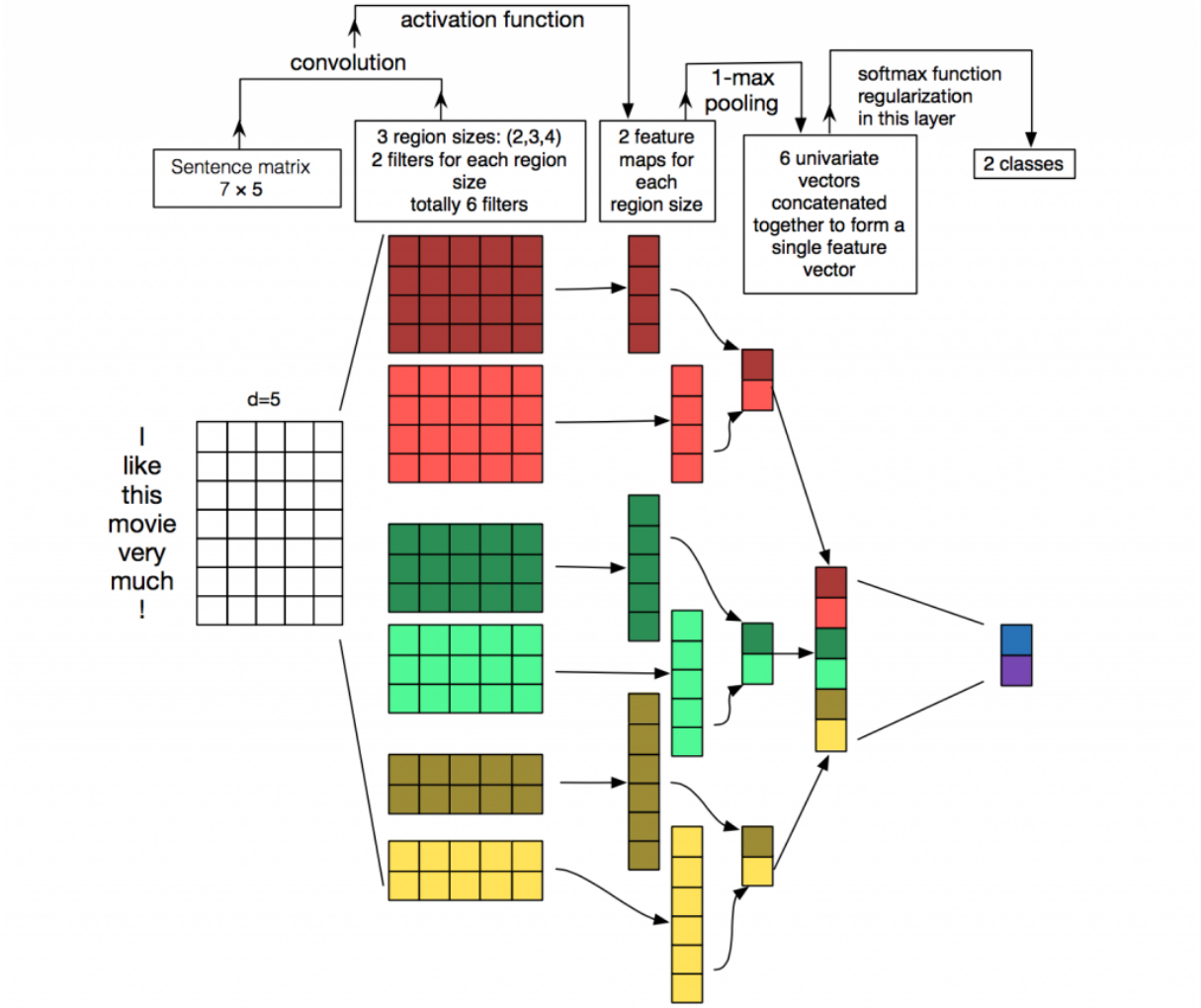

Figure 3.9: Visualizing of a CNN architecture for sentiment analysis.

a vector, CNN forms a "sentence matrix" and use them as the input of model. Those vectors might be taken from the Word2Vec model [23] or Global Vectors for Word Representation model [43]. So that convolutional neural network can capture the distributed representation of words in sentence. A slight variant of CNN model is illustrated in Figure 3.9 [44].

Let $\mathbf{x}_i \in \mathbb{R}^d$ is the d-dimensional word vector representing the i-th word in the sentence. A s-length sentence (padded where necessary) is written as:

$$\mathbf{x}_{1:s} = \mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \cdots \oplus \mathbf{x}_s, \tag{3.25}$$

where $\oplus$ stands for the concatenation operator. We use $\mathbf{x}_{i:i+j}$ to denote the word con-

catenation from $\mathbf{x}_i$ to $\mathbf{x}_{i+j}$. To perform a convolution process, we use a window filter $\mathbf{w} \in \mathbb{R}^{h \times d}$ sliding over h words to generate a new feature. Equation 3.26 shows an example of a feature c_i generating from a sequence of h words $\mathbf{x}_{i:i+h-1}$, from $\mathbf{x}_i$ to $\mathbf{x}_{i+h-1}$:

$$c_i = f(\mathbf{w} \odot \mathbf{x}_{i:i+h-1} + b), \tag{3.26}$$

where $\odot$ represents an element-wise product, $b \in \mathbb{R}$ is the bias value, and f is an activation function, such as tanh, $ReLU$. This window filter will be slided over every possible sequence of words, including $\{\mathbf{x}_{1:h}, \mathbf{x}_{2:h+1}, \ldots, \mathbf{x}_{s-h+1:s}\}$ to generate a *feature map*:

$$\mathbf{c} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{s-h+1}], \tag{3.27}$$

where $\mathbf{c} \in \mathbb{R}^{s-h+1}$. We can refer e feature map as an abstract representation of sentence by using a window filter. In fact, a sentence can be represented by different complementary representations if we investigate different window filters on the same regions. In practice, we use different filters with the same window size, and also with different window sizes. The results of the feature map are generating differently rely on the sentence length and the various filters.

We then apply a max-over-time pooling operation [42], such as *1-max pooling* $\hat{c} = \max\{\mathbf{c}\}$ [45], over each feature map to extract the maximum value from each feature map. The principle is to take the most important information from every feature map. The maximum values of all feature maps are then concatenated to generate a unique feature vector. This new vector will pass through a $softmax$ activation function to get the final result.

Chapter 4

Experiments

This chapter reports results of our experiments to evaluate the deep learning models' performances for sentiment analysis task. We first introduce a new hospitality media dataset crawled from TripAdvisor and do some descriptive analyses in Section 4.1. Secondly, we show some experimental settings of deep learning architectures on Section 4.2. Section 4.3 shows the empirical results and the comparison between these deep leaning models to deepen our understanding about the effectiveness of deep learning models for sentiment analysis task with the new hospitality media dataset.

4.1 Dataset

We used a crawler to obtain a new dataset from TripAdvisor[1], which consists of all English reviews for 410 hotels in Ho Chi Minh City, Vietnam up to February 2017. 58,381 travelers with purchased evidence were giving 75,824 reviews. Among 410 hotels, only 405 hotels had received reviews while the others had not gotten any review yet. Table 4.1 confirms and emphasizes on the upward trend in the number of reviews over years.

Table 4.2 presents the number of hotels, reviews and response rate (%) for each hotel star segment. The final column shows that the higher star of a hotel, the higher of the response rate was given. Moreover, the 3-star to 5-star hotels account for approximate 82% of total reviews. Although 5-star hotels cover only 5.1% in total, they occupied 28.1% reviews in total and actively responded to their customers.

[1]https://tripadvisor.com

Table 4.1: Number of reviews over years. - stands for not available.

	Year													
Star	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2/2017
Unrated	-	4	16	45	119	216	261	253	234	203	134	156	79	2
1	-	-	-	4	5	7	7	10	26	51	153	164	147	1
2	-	3	19	78	232	529	541	819	1,241	1,826	1,958	1,743	1,587	64
3	4	17	51	161	301	1,147	909	1,281	1,766	2,804	3,482	4,498	5,465	215
4	4	20	65	167	166	328	502	842	1,414	2,445	3,090	4,354	5,765	296
5	6	49	112	196	193	331	524	985	1,795	2,771	3,713	4,666	5,738	249

Table 4.2: Number of hotels per star ranking category and its reviews.

Hotel Star	**#Hotels**	**%Hotel**	**#Reviews**	**%Reviews**	**%Response**
Unrated	80	20.2%	1,722	2.3%	15.9%
1	7	1.8%	575	0.8%	8.7%
2	82	20.8%	10,640	14.0%	19.5%
3	160	40.5%	22,101	29.1%	42.6%
4	46	11.6%	19,458	25.7%	65.3%
5	20	5.1%	21,328	28.1%	69.5%

Table 4.3 adds more evidence on the increasing rate of hotel response for the reviews they received from TripAdvisor. Due to the small number of reviews for the years before 2009, we focus on the 2009 afterward. We find that beside the 4-star and 5-star hotels managed to maintain the highest response rate, an increasing trend on the response rate

Table 4.3: Average of response rate (%) over years

		Year							
		2009	2010	2011	2012	2013	2014	2015	2016
Hotel star	1	0	0	0	0	3.9	5.9	7.9	17.7
	2	3.4	4.4	6.2	8.9	16.7	25.4	26.4	36.2
	3	13.7	29.3	20.8	31.2	38.7	44.9	49.6	55.0
	4	3.7	17.3	22.0	42.9	54.0	55.6	78.9	87.7
	5	14.8	16.0	20.0	33.4	57.8	87.7	82.0	86.9

of 1-star to 3-star hotel segments for their customers had also been recorded. As more feedbacks are available to hotels, they are putting more effort on replying their customer's reviews. This basic statistical analysis showed that further study on customer's reviews is needed for hotel manager to effectively facilitate the interaction with their customers.

We visualize the dataset a little more by plotting histograms and box plot of the text length for each star rating using the Seaborn[2] library. The histograms in Figure 4.1 show that the distribution of text length is similar across group 1-star to 2-star ratings, and group 3-star to 5-star ratings. However, the number of customers' reviews seems to be a lot higher towards the 4-star and 5-star ratings.

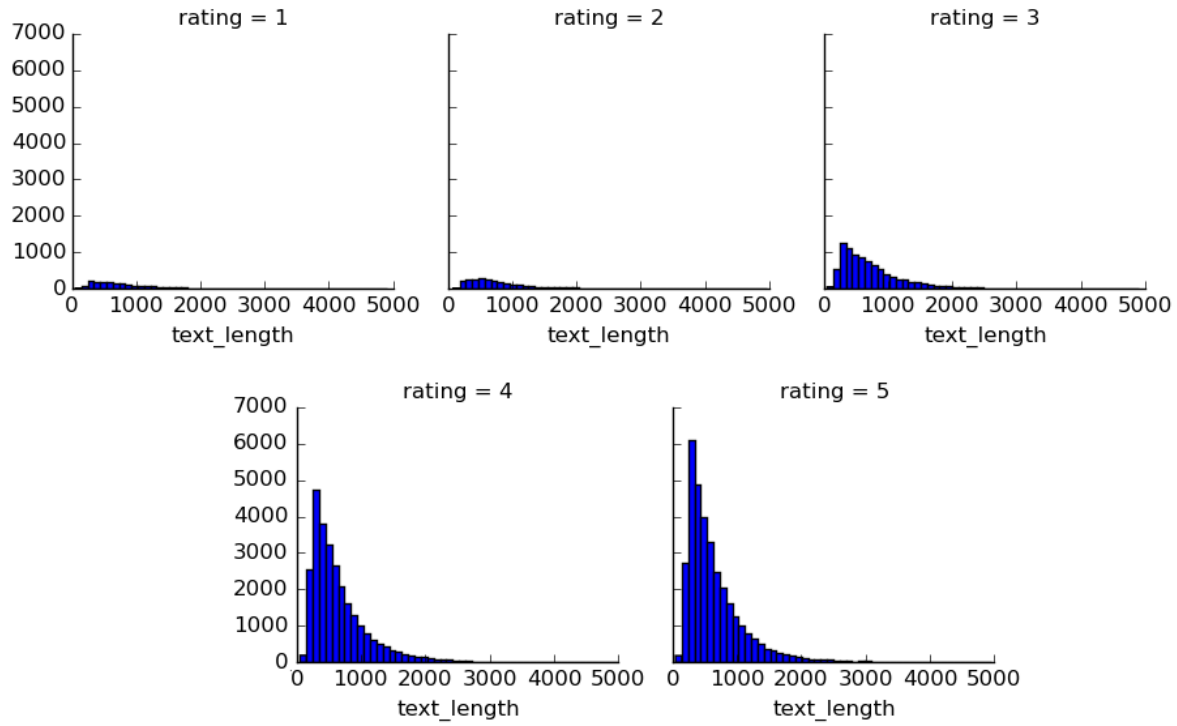

Figure 4.1: Histograms of text length distributions for each star rating.

From the box plot in Figure 4.2, the 1-star to 3-star rating reviews have much longer text compared with 4-star to 5 star rating reviews. That means, the customers expressed their dissatisfaction by writing longer reviews and rated them with low star. However, there are many outliers which can be observed as black points above the boxes. For this reason, we will not consider the text length as a useful feature to determine the sentiment

[2]https://seaborn.pydata.org/

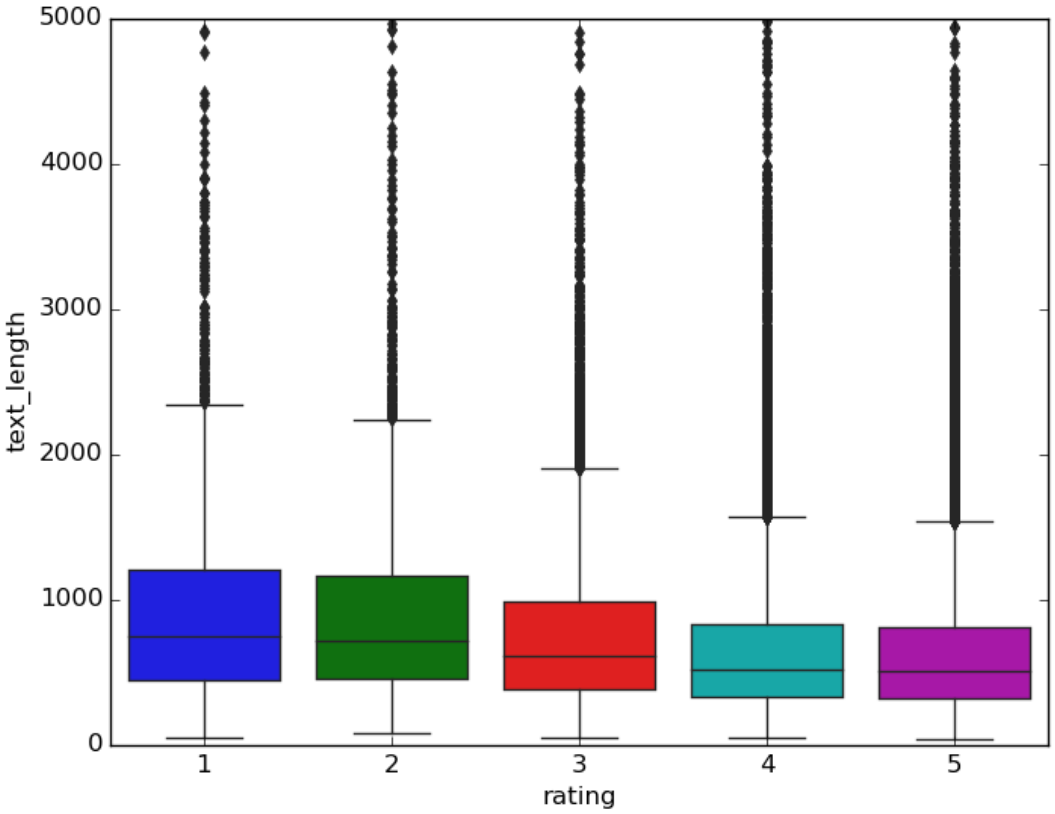

Figure 4.2: Box plot of text length against star ratings.

opinions expressed in hotel reviews after all.

For labeling the hospitality media dataset to train and test the deep leaning models (which can be understood as supervised leaning tasks), we assume that the reviews rated as 4-star and 5-star are positive reviews, while 1-star to 3-star rating reviews consist negative opinions. Table 4.4 shows the details of the hospitality media dataset.

4.2 Experimental Settings

We build our deep learning models using TensorFlow[3], which is a powerful build-in library for implementation of deep leaning algorithms. We employ Adam optimizer [46] due to the advantages that it can archive better results with less training time compared with stochastic gradient descent (SGD). To avoid over-fitting problem, we investigate the

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

Table 4.4: Details of Hospitality media dataset

Hotel star	# Negative	# Positive	% Positive
Unrated	703	1,019	59.18%
1	146	429	74.61%
2	2,003	8,637	81.17%
3	5,746	16,355	74.00%
4	3,321	16,137	82.93%
5	2,420	18,908	88.65%

"dropout" technique [47, 48] at the full-connected layer (non-linear layer), as well as the l_2 norm constraint at softmax loss function (as in [14, 44]). To deal with the different lengths of input sentences, we use zero padding to extend the input dataset to the same length over the sentences. Table 4.5 reveals a list of hyperparameters and their values for each model.

We conduct the experiments using JAIST High Performance Computer (HPC) named UV 3000. We set the same SINGLE node with Intel Xeon E5-4655v3 2.9GHz (6Cores x 2) CPU, 124GB RAM for every model. We use Python[4] 3.6, TensorFlow 1.12 and some build-in libraries for NLP, such as nltk, pandas, sklearn. . . , as the environment for coding and running the experiments.

4.3 Results

We evaluate the performance of these models using a accuracy metric, since it is a classification task. Equation 4.1 describes accuracy metric of the classification task.

$$accuracy = \frac{|X|}{|A|}, \tag{4.1}$$

where X indicates a set of correct labels which are predicted by the model and A is a set of true labels. Obviously, X is a subset of A, i.e. $X \in A$.

We perform 10-fold cross validation on these deep learning models to give an insight of how these models can generalize to an unseen dataset. First, we shuffle the dataset

[4]https://www.python.org/

Table 4.5: Experimental configurations for CNN, LSTM, GRU and BiLSTM

Model	**Description**	**Values**
CNN	input word vectors	Google word2vec
	word embedding size	300
	feature map	(3,4,5), (7,8,9), (7,7,7,7)
	activation function	ReLU
	pooling	1-max pooling
	dropout rate	0.5
	l_2 norm constraint	3
	optimizer	Adam
	mini-batch size	50
	number of epochs	10
LSTM, GRU, BiLSTM	input word vectors	Google word2vec
	word embedding size	300
	hidden units	64, 128, 256
	dropout rate	0.5
	l_2 norm constraint	3
	optimizer	Adam
	mini-batch size	50
	number of epochs	10

randomly, and separate the dataset into 10 equal groups. For each loop iteration, we take a group as test set (10%) and remaining groups as training set (90%). Next, we train these deep learning models with the training set and evaluate them by the test set. We retain the results after discarding the models. Finally, we summarize the performance of these models using the evaluation metric and then calculate the averaged results.

To compare the performance of deep learning models, we also investigate some well-known models as baseline models, which also widely used in sentiment classification task, including Naïve Bayes and Support Vector Machine. These model's architectures are generated as the default model in sklearn[5] library. Table 4.6 shows the performances of

[5]https://scikit-learn.org/

Table 4.6: Results in accuracy and time consuming

Models	**Architectures**	**Accuracy**	**Time consuming**
SVM	Linear	84.02	8818
Naïve Bayes		**83.66**	**8718**
CNNs	(3,4,5)	88.45	11809
	(7,8,9)	88.75	16797
	(7,7,7,7)	88.72	17719
LSTMs	64	88.98	11850
	128	89.06	15901
	256	89.16	29999
GRUs	64	89.02	13632
	128	89.01	19404
	256	89.03	28153
BiLSTMs	64	89.03	15711
	128	89.05	23849
	256	**89.18**	**49611**

deep learning models for sentiment analysis using the new hospitality dataset, both in accuracy and time consuming (model's running time), compared with baseline models.

The first group in Table 4.6 includes the baseline models (Naïve Bayes and SVM), while the rest includes the CNN, LSTM, GRU and BiLSTM models with different configurations. Although Naïve Bayes with default configuration performed the worst accuracy with 83.66 percent on average, it only consumed 8,718 seconds (more than 2 hours) to finish the task, compared with nearly 14 hours of BiLSTMs-256. Comparing the Naïve Bayes with Linear SVM, SVM performed better than Naïve Bayes by 0.36%. However, the result is not statistically significant by computing the Student's t-test at 95% confidence level. It means that the Linear SVM gives no difference to Naïve Bayes in boosting the performance of this task.

On the other hand, the deep learning models archived good performances for the sentiment analysis task in the new hospitality dataset. Comparing with the baseline models, they averagely outperformed Naïve Bayes by 5.3 percent and SVM by 4.9 percent. The

CNN models with different filter region sizes, including (3,4,5), (7,8,9), (7,7,7,7), averagely archived 88.45%, 88.75%, and 88.72% respectively for the sentiment classification task. LSTM models with 64, 128, and 256 hidden units, averagely archived 89.07 percent (89.98%, 89.06%, 89.16% for each model) for the SA task. Furthermore, there is no obvious difference between the performances of GRU models (with 64, 128, 256 hidden units) where the results had been records at around 89.02% on average. Finally, the BiLSTM models averagely performed the best results with 89.09 percent, and the BiLSTM model with 256 hidden units (totally 512 units for both forward and backward state per time step) archived the highest averaged accuracy (89.18%), along with the longest running time (49,611 seconds).

Unexpectedly, the CNN models that archived the state-of-the-art results for sentiment analysis task on document-level were slightly worse than other deep learning models. However, the difference between them is not statistically significant by the Student's t-test at 95% confidence level. That means their performances are comparable for the sentiment analysis task in the new hospitality media dataset.

To deepen our understanding about the performance of the best BiLSTM model with 256 hidden nodes (hereinafter called BiLSTMs-256) for sentiment analysis task in hospitality dataset, we use Tensorboard[6] to visualize and summarize the values of accuracy and loss during training and test phrase of BiLSTMs-256. For deep learning concepts, Accuracy is the common metric for evaluating classification models. As introduced in Equation 4.1, accuracy is the fraction of our model's predictions over the actual sets of labels given by our dataset. At each iteration, we calculate the accuracy of the model at training phrase using training set, and after every 100 iterations, we compute the performance of the model using test set. This is to evaluate how well deep learning model capture the nature of training set, and its adaptation with a new unseen dataset. The accuracy value is shown as numeric numbers between $[0, 1]$ range, which represented for the values between $[0\%, 100\%]$.

Unlike the accuracy, loss values are not a percentage. The loss is calculated on training and test phrases to interpret how well the model is performed for training set and test set. It is clear that the lower of the loss values, the better of the model, unless the model

[6]https://www.tensorflow.org/guide/summaries_and_tensorboard

fall into the over-fitting issue. In other words, loss value is a summation of the errors made for each sample in training or test set. In case of our deep learning networks, the loss function here is the mean squared error classification between the predicted values and the actual values. The main objective in our deep learning models is to minimize the loss value regarding to the model's hyperparameters by changing the weight vector values through Adam optimization method [46]. We also calculate the loss values during each training phrase iteration and every 100 iterations of test phrase using training and test set. Figure 4.3 illustrates the visualization of accuracy value of BiLSTMs-256 model during training and test phrase, while the Figure 4.4 shows the values of loss function of BiLSTMs-256 model during this same period.

During the first 4000 iterations, the accuracy at both training phrase and test phrase steadily increased. This period also witnessed a steady decrease in the loss values at both training and test phrase of BiLSTMs-256 model. The contradiction between those two groups points out that the BiLSTMs-256 model can adapt (or fit) with the training and test set using the Adam optimization method by changing the model's hyperparameters during first 4000 iterations. However, the results during the rest periods showed an opposite trend. The accuracy of model at training phrase gradually fluctuated somewhere near 98 percent, even achieved 100 percent, and its loss values were close to zero. Meanwhile, the mean squared error (loss) values at test phrase significantly increased to 0.100, that led to the fall to under 90 percent of the accuracy at test phrase. The wide gap between accuracy and between loss values at the training and test phrase surely becomes wider and wider if we increase the number of iterations. That means the BiLSTMs-256 model critically captured the nature of training set and too fit with the training data, which negatively impacts the performance of BiLSTMs-256 model on the test data, that led to the over-fitting issue. To overcome this situation, we could reduce the number of iterations by cutting of the number of epochs to three, or taking other techniques into consideration to prevent the over-fitting problem.

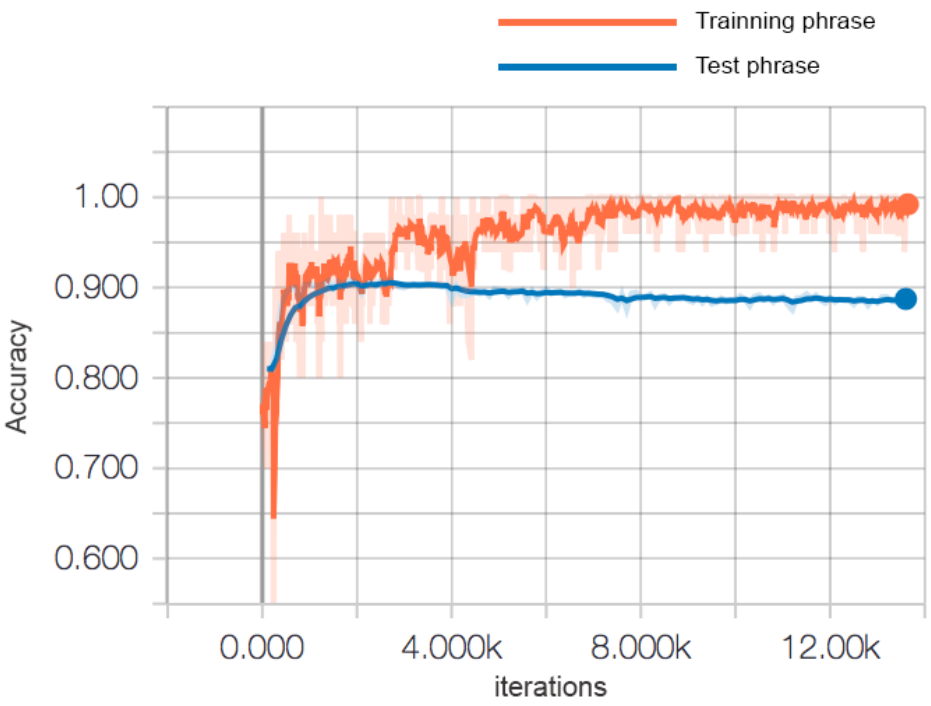

Figure 4.3: Visualizing accuracy of BiLSTMs-256 model during training and test phrase.

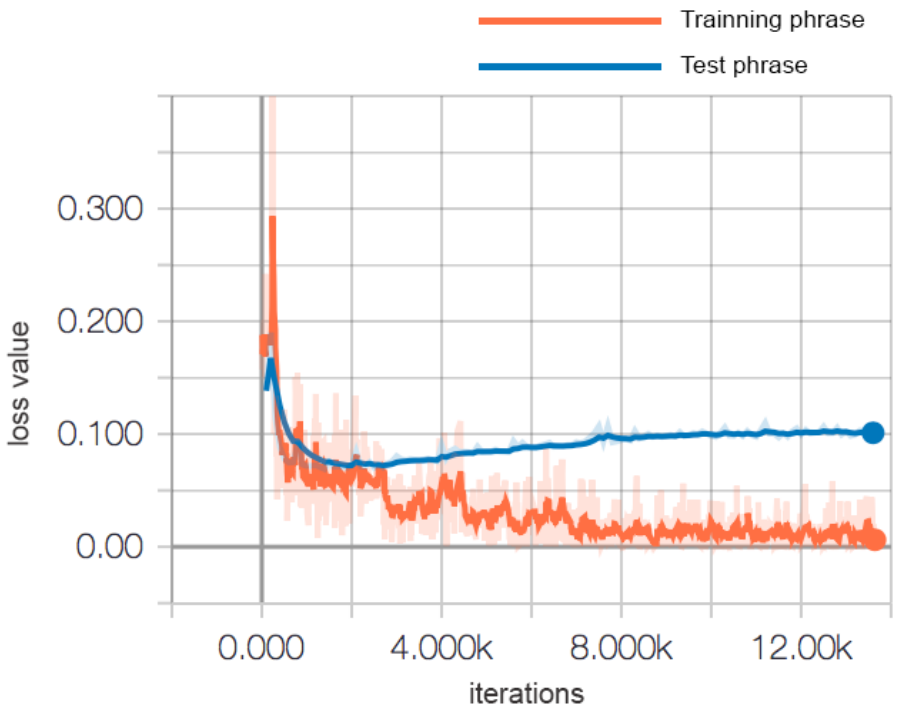

Figure 4.4: Visualizing loss values of BiLSTMs-256 model during training and test phrase.

Chapter 5

Conclusions

5.1 Summary

This thesis conducted a comprehensive survey about sentiment analysis (opinion mining) and current work related to this task, especially using deep learning models to solve the sentiment analysis in hospitality media. We first collected a new hospitality dataset from TripAdvisor which contains all English reviews of 410 hotels in Ho Chi Minh City, Vietnam, and then did some descriptive analyses to understand the nature of this dataset. This fundamental statistical analysis confirmed that all hotel segments (1-to-5 star hotels) highly responded to their customers, and the customer's reviews are the reliable sources for not only hotel managers but also potential customers.

We also examined the architecture of various deep learning models, including Recurrent neural network (RNN), Long-Short Term Memory (LSTM), Gated Recurrent Units (GRU), Bidirectional Long-Short Term Memory (BiLSTM), and Convolutional Neural Network (CNN). These algorithms had been applied and studied in many current studies, and had archived the state-of-the-art results in SA tasks. The major advantage of these techniques is automatically capturing the latent structured representation (abstract representation) of unstructured data in vector space without using any additional knowledge, compared to lexicon-based methods or a feature-based methods. Furthermore, we conducted the experiments using deep learning models for document-level SA in hospitality media. The empirical results of four deep learning models with different architecture settings showed that deep learning techniques could perform well on the new dataset

with comparable results, that had been proved by the Student's t-test at 95% confidence level. These deep learning models overperformed some baseline models, including Support Vector Machine (SVM) and Naiv̈e Bayes, which were well-known in NLP. Finally, the visualization of the best BiLSTMs model with 256 hidden nodes (calculated by the averaged 10-fold cross-validation results) showed that BiLSTMs-256 had been somehow over-fitted. We should take this problem into consideration to boost the performance of deep learning models for sentiment analysis task.

5.2 Limitations

Some limitations of our work should be considered. First, this study only focused on English reviews which might ultimately ignore the significant number of reviews from non-English reviewers. Moreover, the size of this dataset was still limited with only more than 70,000 records. The labeling process must be reconsidered where the positive samples are much larger than negative samples (unbalanced dataset). This feature leads the models tending to predict positive sentiment opinions during the training process and easily leads to the over-fitting problem. Finally, this thesis only focused on document-level sentiment analysis and misunderstood the aspects/topics mentioned in the hotel reviews, which might not include in the user ratings. This information can provide valuable insights to both buy-side and sell-side in the hospitality market to measure the satisfaction of customers and can help businesses improve their products or services. Those limitations can be overcome by taking further research directions in Section 5.3.

5.3 Future Work

In further study, we should pay more attention to aspect-based sentiment analysis. This kind of task is more sophisticated and requires a massive amount of knowledge to identify the specific aspects mentioned in the reviews of a product or service that people were discussing or talking about. Moreover, the further research should focus on the significant number of reviews from non-English reviewers, who also traveled to Ho Chi Minh City and gave the reviews by their mother languages, such as Chinese, French, German, Japanese.... Such kind of reviews could also contribute worthy information to hotel man-

agers to enhance their services and for other travelers who are not familiar with English. Additionally, tracking sentiment over time should also take into consideration the seasonal effects especially in the hospitality industry, so that our study could be extended to monthly and seasonally, rather than annually, review sentiments.

Bibliography

[1] B. Liu, *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, 2012.

[2] K. H. Yoo and U. Gretzel, "What motivates consumers to write online travel reviews?," *Information Technology & Tourism*, vol. 10, no. 4, pp. 283–295, 2008.

[3] S. Schmunk, W. Höpken, M. Fuchs, and M. Lexhagen, "Sentiment analysis: Extracting decision-relevant knowledge from ugc," in *Information and Communication Technologies in Tourism 2014*, pp. 253–265, Springer, 2013.

[4] J. A. Chevalier and D. Mayzlin, "The effect of word of mouth on sales: Online book reviews," *Journal of marketing research*, vol. 43, no. 3, pp. 345–354, 2006.

[5] H. Khang, E.-J. Ki, and L. Ye, "Social media research in advertising, communication, marketing, and public relations, 1997–2010," *Journalism & Mass Communication Quarterly*, vol. 89, no. 2, pp. 279–298, 2012.

[6] B. Liu and L. Zhang, "A survey of opinion mining and sentiment analysis," in *Mining text data*, pp. 415–463, Springer, 2012.

[7] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 347–354, 2005.

[8] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, 2014.

[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning.," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[11] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," in *Proceedings of Advances in Neural Information Processing Systems*, pp. 1799–1807, 2014.

[12] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[13] T. N. Sainath, A. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8614–8618, 2013.

[14] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751, 2014.

[15] H. Nguyen and K. Shirai, "A joint model of term extraction and polarity classification for aspect-based sentiment analysis," in *Proceedings of the 10th International Conference on Knowledge and Systems Engineering (KSE)*, pp. 323–328, 2018.

[16] A. Bordes, S. Chopra, and J. Weston, "Question answering with subgraph embeddings," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 615–620, 2014.

[17] S. Jean, K. Cho, R. Memisevic, and Y. Bengio, "On using very large target vocabulary for neural machine translation," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, pp. 1–10, 2015.

[18] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2*, pp. 3104–3112, MIT Press, 2014.

[19] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate.," *CoRR*, vol. abs/1409.0473, 2014.

[20] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization.," *CoRR*, vol. abs/1509.00685, 2015.

[21] M. A. Nielsen, *Neural networks and deep learning*, vol. 25. Determination press USA, 2015.

[22] H.-X. Shi and X. Li, "A sentiment analysis model for hotel reviews based on supervised learning," in *Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC)*, vol. 3, pp. 950–954, 2011.

[23] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of Advances in Neural Information Processing Systems*, pp. 3111–3119, 2013.

[24] R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit," *Nature*, vol. 405, no. 6789, pp. 947–951, 2000.

[25] S. Hochreiter, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," *A Field Guide to Dynamical Recurrent Neural Networks*, 2001.

[26] S. Hochreiter, J. Urgen Schmidhuber, and C. Elvezia, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[27] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proceeding of the Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[28] H. Zhiheng, X. Wei, and Y. Kai, "Bidirectional lstm-crf models for sequence tagging," *CoRR*, vol. abs/1508.01991, 2015.

[29] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. 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*, pp. 260–270, 2016.

[30] P. Chen, Z. Sun, L. Bing, and W. Yang, "Recurrent attention network on memory for aspect sentiment analysis," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 452–461, 2017.

[31] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, 2014.

[32] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm networks," in *Proceedings of the 2005 International Joint Conference on Neural Networks*, 2005.

[33] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proceedings of NAACL-HLT*, pp. 260–270, 2016.

[34] T. Chen, R. Xu, Y. He, and X. Wang, "Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN," *Expert Systems with Applications*, vol. 72, pp. 221–230, 2017.

[35] M. Liwicki, A. Graves, S. Fernàndez, H. Bunke, and J. Schmidhuber, "A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks," in *Proceedings of the 9th International Conference on Document Analysis and Recognition, ICDAR*, 2007.

[36] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[37] P. Wang, J. Xu, B. Xu, C. Liu, H. Zhang, F. Wang, and H. Hao, "Semantic clustering and convolutional neural network for short text categorization," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, vol. 2, pp. 352–357, 2015.

[38] Y. Zhang, S. Roller, and B. C. Wallace, "Mgnc-cnn: A simple approach to exploiting multiple word embeddings for sentence classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1522–1527, 2016.

[39] W. Yih, X. He, and C. Meek, "Semantic parsing for single-relation question answering," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, pp. 643–648, 2014.

[40] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "Learning semantic representations using convolutional neural networks for web search," in *Proceedings of the 23rd International Conference on World Wide Web*, pp. 373–374, ACM, 2014.

[41] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, pp. 655–665, 2014.

[42] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[43] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[44] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners guide to) convolutional neural networks for sentence classification," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, pp. 253–263, 2017.

[45] Y. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 111–118, 2010.

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization.," *CoRR*, vol. abs/1412.6980, 2014.

[47] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. abs/1207.0580, 2012.

[48] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

Publications

International conferences

[1] Thang Tran, Hung Ba, and Van-Nam Huynh, Measuring Hotel Review Sentiment: An Aspect-Based Sentiment Analysis Approach. *In: Proceedings of the 7th International Symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making (IUKM)*. March $27^{th} - 29^{th}$ 2019, Nara, Japan. *Accepted*.