Title	A STUDY OF ASPECT-BASED SENTIMENT ANALYSIS FOR ONLINE FOOD DELIVERY PLATFORMS
Author(s)	張, 子涵
Citation	
Issue Date	2022-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/17730
Rights	
Description	Supervisor:Van-Nam HUYNH, 先端科学技術研究科, 修士（知識科学）

Master's Thesis

A STUDY OF ASPECT-BASED SENTIMENT ANALYSIS FOR ONLINE FOOD DELIVERY PLATFORMS

1910422 Zhang Zihan

Supervisor: Professor HUYNH, Van Nam

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
(Knowledge Science)

January 2022

Abstract

Nowadays, with the rapid development and innovation of new technologies, the Internet has become the main driving force of national and even global economic growth, which has greatly enriched people's lives. Millions of users can obtain information, exchange information, express their views and share their experiences, and express various emotions on the Internet. When people want to evaluate something, they often have their subjective emotional tendency, and when people want to make decisions, they usually refer to other people's opinions. From then on, people's lives are more and more affected by the information on the Internet. Therefore, sentiment analysis and applications have become more critical and popular as a branch of Natural Language Processing (NLP).

This thesis studies on the online food delivery platforms that have comprehensive coverage of loyal user groups among many social media platforms. Analyzes the user's emotion behind the text through the massive comments data generated by the on-demand delivery apps and clarifies the user's attitude towards service and orders, which is of positive significance to the improvement and development of providing services. The thesis includes two main parts. The first part summarizes the related work of sentiment analysis and the basic concepts of Deep learning models. The second part experiments the sentiment analysis for the comment data on the online food delivery platforms as the research object to explore its potential user sentiment. We used the deep learning language model to realize the short text sentiment classification task compared with traditional practical analysis.

In addition, this thesis applies the transfer learning technique by using the pre-training BERT model. Then, the fine-tuning process updates the pre-training parameters of the model. Finally, the corresponding online food delivery platforms data set is used to evaluate the performance of the trained model. The results show that the above improvement can achieve classification accuracy in the final sentiment classification stage. It also shows the progress of fine-grained sentiment analysis on online food delivery services, directing further research in this field.

Keywords: Online food delivery, sentiment analysis, natural language processing, BERT model

List of Figures

List of Tables

Contents

[12pt,a4paper,openright]book

Chapter 1

Introduction

1.1 Online food delivery platform

Online services have been attracting researchers' attention and food delivery applications for a long time. In fact that, users could not only score the star of the merchant (1-5 star), but also comment on the merchant's services or products in details. In the follow-up study, we made a statistical analysis on the comments of the Meituan merchants, which is the Chinese largest online food delivery platform, in June 2021. The distribution of users' emotional tendency among different star-rated merchants are analyzed. The overall star rating of merchants reflects the users' overall sentiment tendency towards the merchant. At the same time, we found that there is a lot of negative emotional content in the comments of high star rated merchants. There are also many positive emotional contents in the comments of low star merchants. The reason is that a large number of star ratings in user comments are inconsistent with the emotional tendency of corresponding comments. There are negative emotional content in high star comments and positive emotional content in low star comments.

With the development of transportation and online services, especially, the impact of COVID-19, many conventional restaurants that only focus on dine-in service have been forced to close, while the online food delivery service has proliferated. As shown in Figure 1.1, the figure provides the variable annual growth rate of 9 cities in China from 2019 to 2020 which shows that delivery service contributed to the sales of restaurants during the COVID-19 pandemic [10].

On one hand, when a service comes to online ordering, users are often referring the reviews or comments from others for making their own decisions. For example, when the people are considering about where to go, what to eat, and what time is suitable? especially, when users are in a new place, when they are uncertainty about the surrounding environment. On the other hand, the providers or restaurants want to understand more about their customers. Then, they will improve the service to add more values

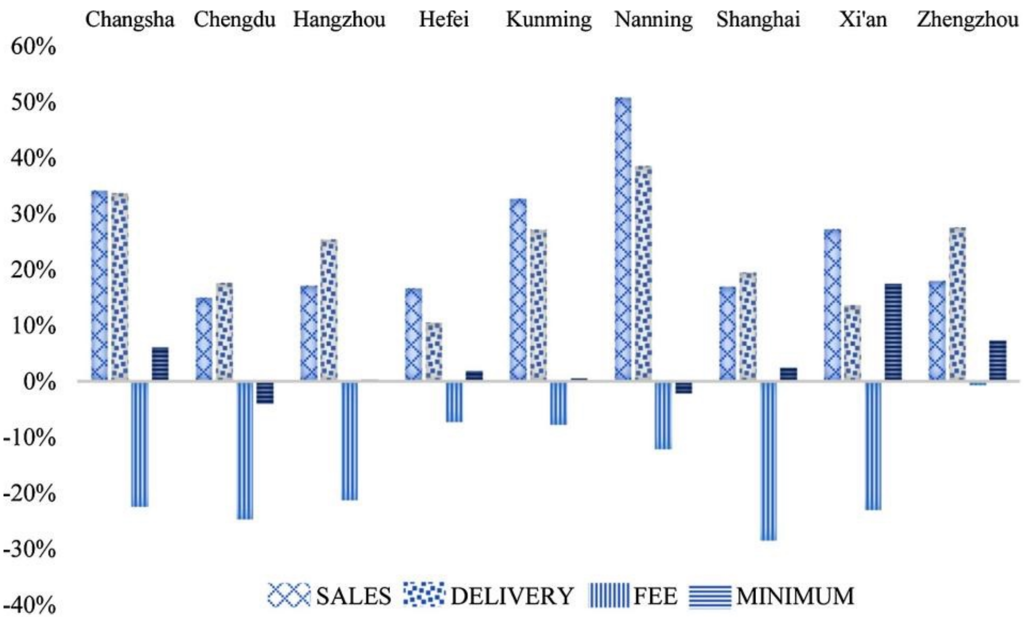


Figure 1.1: Annual growth in attributes by cities (Taken from Kim et al, 2021).

to the users. Therefore, the providers must know more feedback from their customers by reading the comments or mining the opinions from the experiments of the users.

However, it is challenging to monitor the enormous numbers of comments on the websites manually. Additionally, different experts have different opinions, and that can easily cause biases in practical work. In recent years, to handle these situations, researchers have explored a new research direction in Natural Language Processing (NLP) [11,12] named aspect-based sentiment analysis (ABSA) [13, 14]. It considers both the sentiment and aspect information. Aspect-based sentiment analysis contains two sub-tasks: aspect extraction and aspect sentiment analysis. The former extracts aspects from the input, and the latter classifies the sentiment expressed on those aspects.

1.2 Study questions

This thesis aims to answer to following questions under perspectives of knowing the insights of the business model, then using data analysis techniques, and deep learning models to support the online food delivery services.

- How do the online food delivery platforms run?

- How can we use the comment sentences or textual dataset on the platform to mine opinions from the customers? Therefore, from viewpoint of service providers who will add more values to the customers, and from view point of customers who will have the references from others for making decisions.
- What is the sentiment analysis, and are the sentiment analysis challenges for the online food delivery services?
- How to use natural language processing techniques for solving those sentiment analysis challenges?
- How to collect the comment datasets from the platforms, and preprocess these datasets?
- How to analyze and discuss about the experimental results?

1.3 Study aims and future direction

This study achieved the following main points:

- Fist, we understand the insights of the online food delivery business model. For example, the necessary of the online delivery platforms due to the demand from the customers, especially for young people. Who want to use high technology for their routing daily. We know the important of the customer comments for running the online food delivery platforms.
- We did the experimental test for sentiment analysis tasks on the textual dataset. We know the pipe-line of the data analysis procedure. Then, we believe in using the data analysis techniques not only for sentiment analysis on the online food delivery platforms, but also for many other kind of online services or the recommendation systems in general.

We hope that through this study, on the one hand, we can fill the gap in this research field and deepen our understanding of aspect-based sentiment analysis, on the other hand, we can improve the economic effect of relevant platforms and customers' ordering experience.

In future works, on one hand, we will continue to experiment more on the sentiment analysis tasks on the other application domains. For example, we will mine the opinions from difference social platforms. Then, we can obtain the insights of the customer more clearly by combining more sources of the information. On the other hand, we also want to improve the techniques that apply for the sentiment analysis tasks. For example, within the rapid development of the deep learning models, how we can use the transfer learning to shorter time for training a new model. Then, it is possible to mine the opinions or ideas from users in real-time approximately.

Chapter 2

Background

2.1 Sentiment analysis

Sentimental analysis concept [15, 16] also known as opinion mining, is a research problem that analyze people's opinion, sentimental, appraisal, attitude, and emotion about an entity expressed through text. The entities here can be products, services, organizations, individuals, events, and various topics. Sentiment analysis can generally be divided into three levels: document level, sentence level, and aspect level.

Document level divides the sentiment of the whole document into positive or negative. This can be seen as a binary classification problem (it is unlikely that there are comments without emotion because the purpose of writing comments is to express feeling, even fragile emotion), so it is also called document-level sentiment classification. For example, for a comment on a commodity, we can analyze whether the sentiment of the comment is positive or negative overall. It assumes that all comment sentences in this document are directed to the same entity. If a document will comment on multiple entities, this level of analysis is problematic.

Sentence level determines whether its sentiment is positive, negative, or neutral for each document sentence. Unlike the previous document level, some descriptive sentences have no sentiment, so there is a neutral classification here, which means no sentiment. This problem has something to do with subjective and objective classification. This task is to judge whether a sentence is subjective or objective. Usually, neutral sentences are objective, while positive or negative sentences are subjective. But they are not the same. For example, “We should have the car last month, and the windshield wiper has fallen off” is an objective sentence, but it describes an undesirable thing, so it implies negative sentiment. Although the sentence “I think he went home after lunch” is subjective, it has no positive or negative sentiment.

Aspect level is different from both the document level and sentence level. It needs to consider the target information (attribute) and its corresponding emotion simultaneously. For example,"this restaurant is well decorated, the food is also delicious, but the location is very inconspicuous, it took me a long time to find it, and the waiter's attitude is not good." Overall, it is difficult to decide whether this sentence should be judged as positive or negative because in the comments, users praised the decoration and dish attributes of the restaurant but felt that the location and service attributes of the restaurant were not good. We need to identify aspect-level sentiment with finer granularity to get more comprehensive and accurate sentiment information.

The aspect level analysis is academically called ABSA (aspect-based sentimental analysis) [17,18], which can be divided into ACSA (aspect category sentimental analysis) and ATSA (aspect term sentimental analysis). The ACSA is to identify the sentiment tendency in the corresponding predefined attribute category (aspect category). For example, the above comments are in the attribute category "dish taste". Positive sentiment is expressed on the attribute category "service attitude", and negative sentiment is expressed on the attribute category "service attitude". The ATSA identifies the text's sentiment tendency for the corresponding attribute (aspect term). For example, the above comments express positive sentiment for the attribute "dish" and negative sentiment for the attribute "waiter". The aspect level sentiment analysis described in this paper mainly refers to the ACSA task.

2.2 Long-Short Term Memory

Long short term memory (LSTM) is an improved version of recurrent neural network (RNN) [19]. The RNN model is a typical back propagation recurrent neural network, in which back propagation (BP) [20] is also called error back propagation. The RNN is very effective for the data with sequence characteristics. It can mine the temporal information and semantic information in the data. Using this ability of RNN, the deep learning model has made a breakthrough in solving many problems in NLP fields.

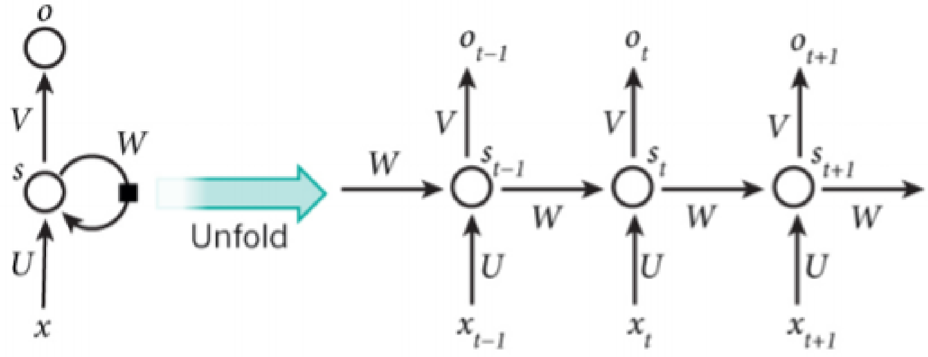


Figure 2.1: RNN structure [1]

Figure 2.1 shows the structure of the RNN after expanding it into a complete network. The meaning expressed here is the network structure of the whole sequence. The $t-1$, t, and $t+1$ represents the duration of the time series x. The input sample $S_t = f(W * S_{t-1} + U * x_t)$ represents the memory of the sample at time t, W represents the weight of the input, U represents the weight of the input sample at the moment, and V represents the weight of the output sample. When $t = 1$, input $S_0 = 0$ for general initialization, randomly initialize W, U, and V, and calculate with the following formula:

$$h_1 = U_{x_1} + W_{s_0} \tag{2.1}$$

$$s_2 = f(h_1) \tag{2.2}$$

$$O_1 = g(V_{s_1}) \tag{2.3}$$

Where f and g here are activation functions, f can be tanh, relu, sigmoid and other activation functions, and g is usually softmax. The W, U, and V are equal at each time, which means, the weight is shared. We can see that the hidden state at each time is not only determined by the input at that time but also depends on the value of the hidden layer at the previous time. If a sentence is very long, it will not remember the beginning and details of the sentence at the end of the sentence. Although the RNN has a good effect on time series problems, there are some problems such as gradient disappearance or gradient explosion due to the long-term dependence of the BP algorithm on time series. The improved model LSTM based on this is one of the most successful methods. Schmidhuber [21] proposed the LSTM model in 1997.

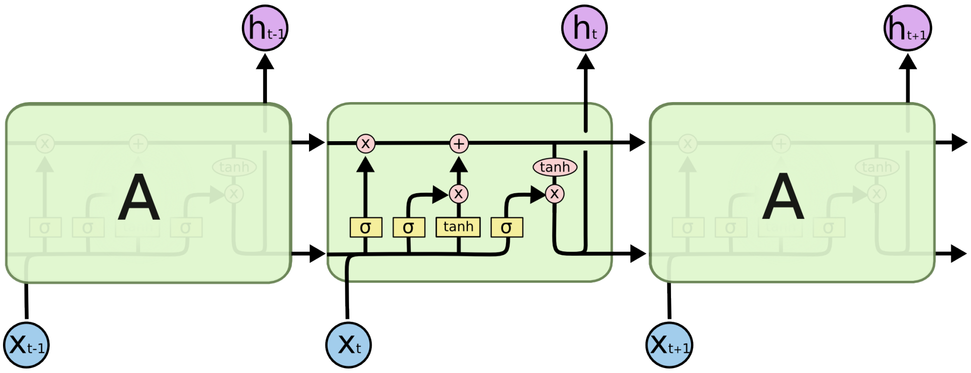


Figure 2.2: LSTM structure [2]

The difference between RNN and LSTM is that $h_t = U_{x_t} + W_{s_{t-1}}$ in RNN is a simple linear summation process. As shown in Figure 2.2, LSTM can remove or increase the information of “cell state” through the “gate” structure to realize the retention of important contents and the removal of unimportant contents. The sigmoid layer outputs a probability value between 0 and 1 to describe each part that can pass through. 0 means “no variables are allowed to pass through”, and 1 means “all variables are allowed to pass through”. The switch for forgetting information is called the forgetting gate, the switch for adding information is called updating gate, and the switch for outputting results is called the output gate. LSTM is specifically designed to avoid long-term dependency problems. They are effective on various issues and are now widely used in the field of deep learning.

2.3 Word2vec technique

Since Mikolov [22] put forward the concept of word vector in his 2013 paper “Effective estimation of word representation in vector space”, the NLP field seems to have suddenly entered the embedded world, sentence2vec, doc2vec, and everything2vec. Based on the assumption of the language model - “The meaning of a word can be inferred from its context”, the word vector puts forward the distributed representation of words, which is compared with the high-dimensional and sparse representation of traditional NLP, the word vector trained by word2vec is low dimensional and dense. Word2vec uses the context information of words to enrich the semantic information. At present, common applications include:

- Use the trained word vector as the input feature to improve the existing systems, such as the input layer applied in neural networks such as emotion analysis, part of speech tagging, language translation, and so on.

- Apply word vectors directly from the perspective of linguistics, such as using the distance of vectors to express word similarity, query relevance, etc.

Before the birth of word2vec, there was no unified method to represent text in the NLP field, although there were some special methods to represent text, such as the one-hot vector or bow method. Chinese text segmentation usually adopts various sequence annotation methods and then segments the text according to semantics. For example, k-singles cuts a text into some text segments into Chinese and use various sequence annotation methods to segment the text according to semantics.

The TF-IDF [23] technique uses frequency to represent the importance of words. In-text rank, the page rank method is used to describe the weight of words. The LSA [24] is based on SVD pure mathematical decomposition word document matrix. P in LSA, the document formation process is characterized by probability means, and the solution result of the word document matrix is given probability meaning. Two conjugate distributions were proposed in LDA [25, 26] to introduce a prior and so on perfectly.

Word2vec is a process of using a one-layer neural network (i.e., CBOW) to call the sparse word vector mapping in the one-hot form an n-dimensional (n is generally hundreds of) dense vector. In order to speed up model training, the tricks include hierarchical SoftMax, negative sampling, Huffman tree, etc.

In NLP field, the most fine-grained object is words. If we want to label parts of speech, we can have a series of sample data (x, y) with a general idea. Where x represents words and y represents parts of speech. What we need to do is to find a mapping relationship between $x \to y$. We can apply traditional methods including Bayes, SVM, and other algorithms [27]. However, our mathematical models are generally numerical inputs.

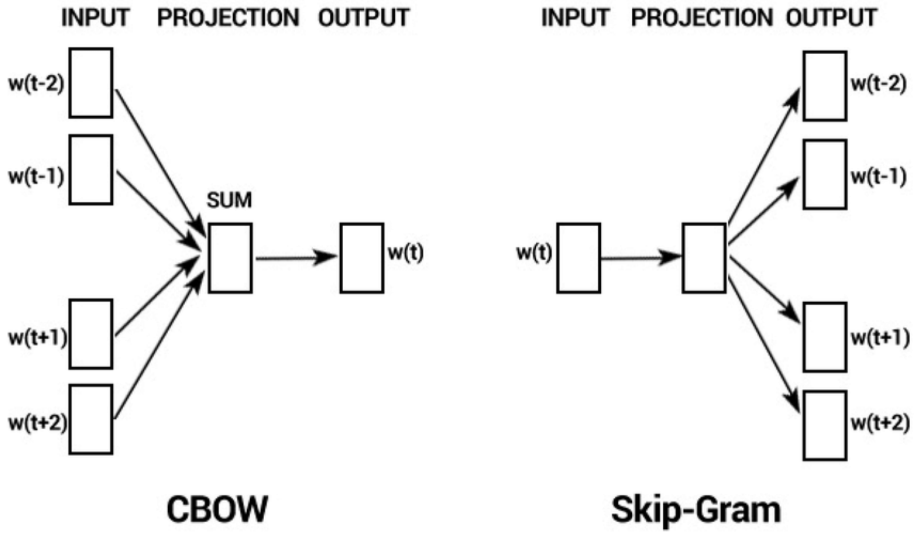


Figure 2.3: CBOW model and Skip-gram model [3]

In fact, the words in NLP are the abstract summary of human beings. They are in symbolic form (such as Chinese, English, Latin, etc.), so they need to be converted into a numerical form or embedded into a mathematical space. This embedding method is called word embedding, and word2vec is a kind of word embedding. For example, x is regarded as a word in a sentence, y is the word's contextual word. Then function here is the “language model” that often appears in NLP. The purpose of this model is to judge whether the sample (x, y) conforms to the laws of natural language. More commonly, word x and word y together to see whether the sentence has sense. Word2vec is derived from this idea, but its ultimate goal is not to train function perfectly, but to only care about the model parameters (especially the weight of neural network) as the by-product of model training and take these parameters as a vectorized representation of input x, which is called word vector. There are two essential models in word2vec - CBOW model (continuous bag of words model) and Skip-gram model. The schematic diagram is given in Tomas mikolov's [22] paper.

It can be seen from the name and the illustrated Figure 2.3 that CBOW is to calculate the probability of a word according to the “C” words in front of a word or the “C” consecutive words before and after a word (“C” stands for the number of words). On the contrary, Skip-Gram model is based on a particular word and then calculates the probabilities of certain words before and after it.

2.4 Attention architecture

In short, when we are looking at something, the current focus must be on a specific place of what we are looking at. When our line-of-sight shifts, our attention shifts with the change in the line of sight, which means that when people pay attention to a target or a scene, the distribution of attention in each spatial position of the target and scene is different. For the comment text, the influence of each word in the sentence on its final sentiment classification is distinct. To expand the power of the critical parts, it is necessary to find and highlight the key features.

Therefore, based on the two-way long-term and short-term memory network, an attention mechanism is introduced to extract the relatively essential parts of the text for emotion classification and improve its weight in the final generated text features. The greater the weight, the more critical the representative is for emotional polarity classification. Hence, predecessors have proposed the attention model, and the formula is as follows:

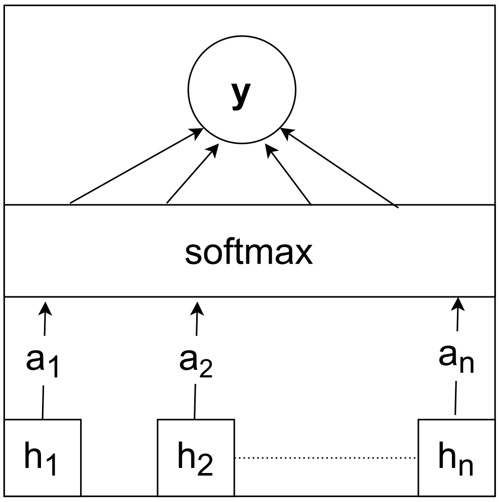


Figure 2.4: Attention structure adapted fromJun 24,2018. [4]

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} \times h_j \tag{2.4}$$

where c_i represents the attention of i, T_x represents the sentence length, α_{ij} represents the attention distribution coefficient of the j_{th} word in the source input sentence when the i_{th} word is output by target, and h_j is the semantic code of the j_{th} word in the source input sentence. The calculation of attention formula will be introduced in more details in Chapter 3.

2.5 Chinese word segmentation

As shown in Figure 2.5, the basic algorithms of Chinese word segmentation mainly include dictionary-based methods, statistics-based methods, rule-based methods, and neural network-based methods. According to the current research, the method based on Neural Network combined with machine learning is the best. Then first briefly explain the development of the Chinese word segmentation algorithm, and then introduce the neural network word segmentation method used in this paper.

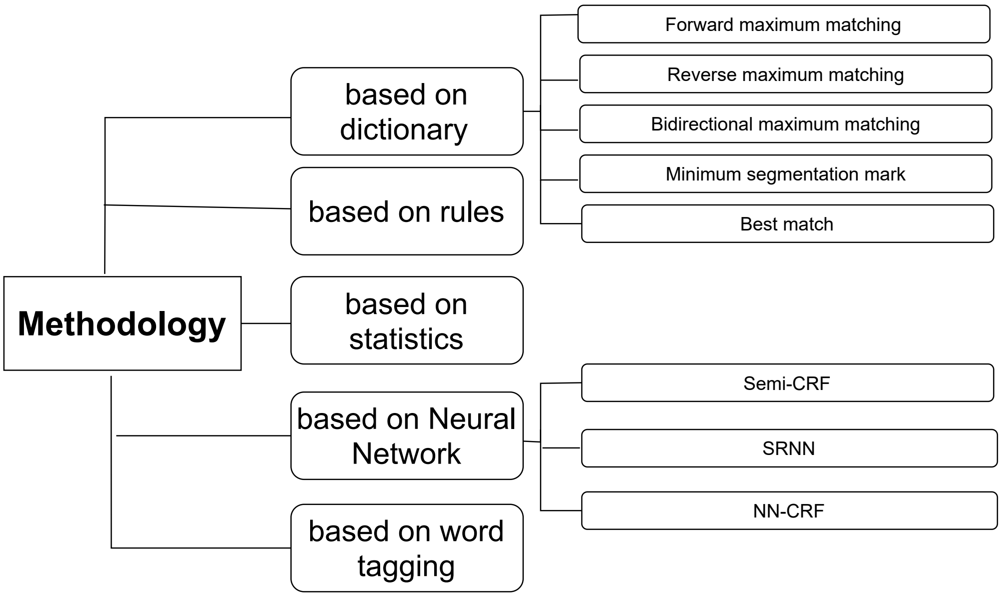


Figure 2.5: Word segmentation.

Figure 2.6 introduces the general flow of forward and reverse maximum matching algorithm. The bidirectional maximum matching algorithm compares the word segmentation results obtained by the forward maximum matching method with the results obtained by the reverse maximum matching method to select the correct word segmentation method. According to the research, 90% of Chinese sentences are entirely consistent with the results obtained by the forward maximum matching algorithm and the reverse matching algorithm. In the remaining 10%, 90% of the two algorithms must be correct, and only 1% of the sentences is divided incorrectly using the maximum matching. Therefore, thanks to its strong word segmentation ability, the maximum matching algorithm has existed for a very long time in the history of word segmentation.

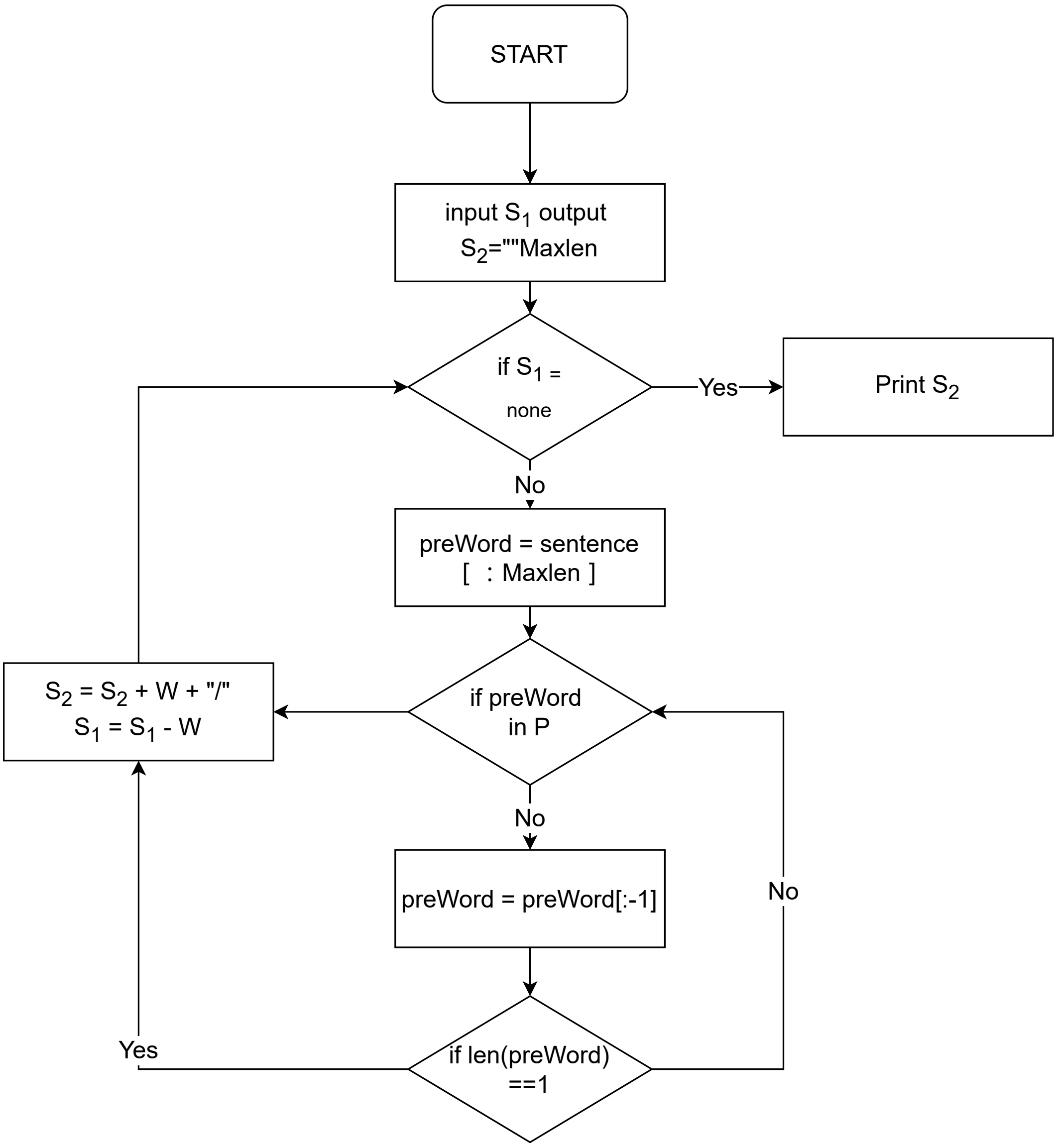


Figure 2.6: Forward and backward maximum matching algorithm. [5]

Statistics-based word segmentation is another major category of word segmentation task implementation. As the name suggests, statistics-based means that the more times a word appears in the context, the more likely these words are to form a word. The typical algorithms are N-gram and the Hidden Markov model (HMM) [28]. The idea of N-gram requires that the probability of the occurrence of a word in a sentence is only related to all the words in front of it, not to the words in the rear relative position. The probability of the occurrence of the whole sentence is just the product of the probability of the occurrence of each word constituting the sentence.

The method of the neural network is described as follows. In 2016, Kong et al. [29], proposed a new algorithm, SRNN (partial recurrent neural networks), which combines semi-CRF (semi conditional random field) [30] and neural network. It is not difficult to find that semi-CRF originally came from the conditional random field (CRF). The conditional random field is a Markov process-based modeling, in which each element of the input sequence is marked at each step of the random process. Semi means half, so semi-CRF is a semi-Markov-based modeling process; all continuous elements in the input sequence are marked at each algorithm step. The word segmentation task has natural advantages. When marking constant and identical tags, it can recognize words from the input sequence. The specific - calculation formula is as follows:

$$p(s|x) = \frac{1}{Z(x)} \times \exp\{W \times G(x,s)\} \tag{2.5}$$

where x represents the input sentence sequence, s represents the corresponding word segmentation sequence, and $G(x,s)$ is the mapping function that converts x and s into feature vectors. In the SRNN model, the author hopes to use a bidirectional RNN to model the G function. The bidirectional RNN can combine the input word sequence vectors into word vectors. The framework of the model is shown in Figure 2.7.

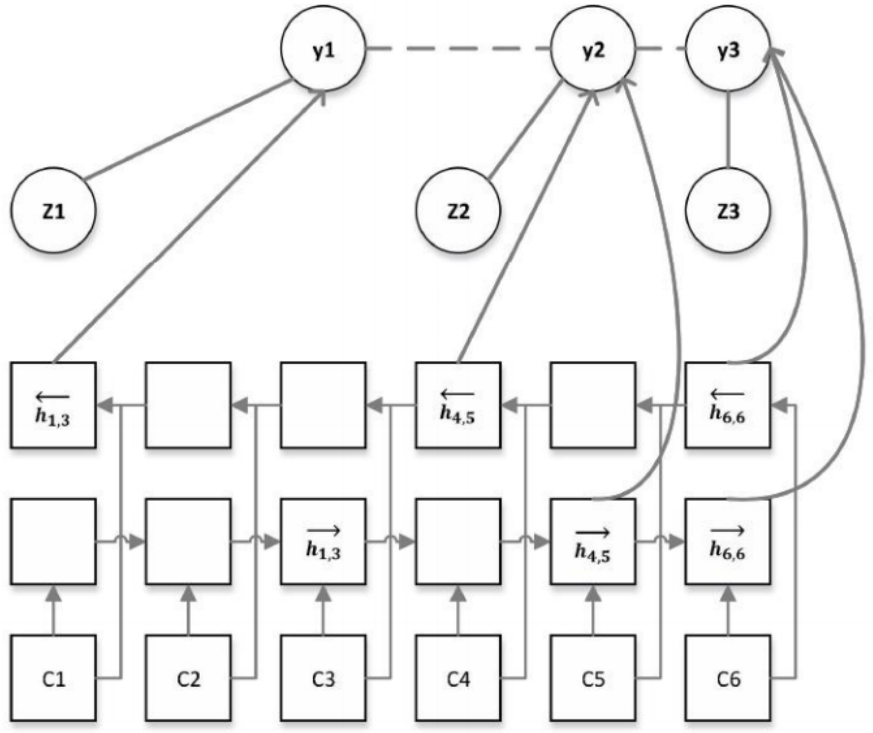

y1
y2
y3
Z1
Z2
Z3
$\overleftarrow{h}_{1,3}$
$\overleftarrow{h}_{4,5}$
$\overleftarrow{h}_{6,6}$
$\overrightarrow{h}_{1,3}$
$\overrightarrow{h}_{4,5}$
$\overrightarrow{h}_{6,6}$
C1
C2
C3
C4
C5
C6

Figure 2.7: SRNN structure adapted from Biswas, S., Gall, J. (2018, March) [6]

According to the previous research results, the key to the problem is constructing the G function. If a better neural network is designed to construct the G function, the effect of word segmentation may be effectively improved. With the help of this idea, Zhuo et al [31], started from the combined network; in addition to taking the network structure as the G function, they also added the word vector representation to the G function. The network structure is shown in Figure 2.8.

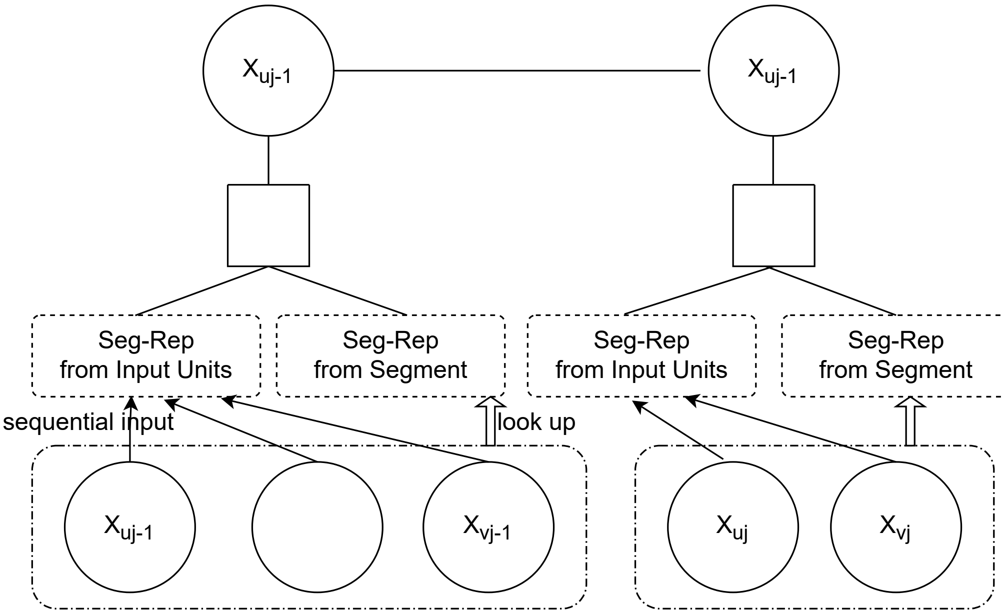


Figure 2.8: Combining word vectors and neural networks adapted from Rong, X. (2014). [7]

Among them, “Seg-Rep from input units” represents SRNN, and “Seg-Rep from the segment” represents some word vectors of the input sequence. In this model, the large-scale input text is segmented using the word vector independent model SRNN; the Word2vec word vector learns the segmented words. The vector output is used as the input of the G function, and finally, the word segmentation result is obtained. The word segmentation effect on the public data set is significantly improved using this method. After the online food delivery platforms comment text data set is segmented using the method shown in Figure 2.7, the pre-training model provided by BERT can be used for pre-training. To shorten the training time, we use the training parameters provided by Google. The specific training details and the following fine-tune method are in detail in Chapter 3.

Chapter 3

BERT model

3.1 Introduction to BERT model

Converting natural human language into understandable computer representation has been a research hotspot for a long time. For example, the typical methods such as One-hot, Word2vec, and GloVe proposed by Pennington et [32] are based on the solution of this problem. The main purpose of these methods is to learn context-free information representation.

In recent years, the academic community has taken the representation of learning context-related information as the direction of technological breakthroughs. ELMo uses the Bidirectional LSTM model to learn context-related information; CoVe proposed by McCann et al. [33] uses machine translation to embed context information into word representation.

This section introduces the BERT model to implement word embedding in context. BERT is pre-trained based on the Markup Language Model and Bidirectional Transformer. BERT is the first representation model based on fine-tuning. Inspired by the previous pre-training, Bert also adopts the model structure of pre-training and fine-tuning. The pre-training model mainly adopts two training methods, Masked Language Model (MLM) and Next Sentence Prediction (NSP). The specific methods of these two pre-training tasks will be introduced in detail below. Then, we will introduce the preparation of the BERT model.

The model's input can clearly represent a sentence or a pair of sentences in the tag sequence, such as the [question, answer] pattern. It should be noted that the sentences mentioned here can be continuous texts of any span, rather than semantically complete sentences in our real expression. Sequence refers to the sequence label input into the model. It can be one or two sentences combined into a composite sentence. The input representation consists of the sum of Token Embeddings, Segment Embeddings, and Position Embeddings [34].

Figure 3.1 is a logical description of the input representation. The plain text content after data cleaning is input through input. First, use Wordpiece

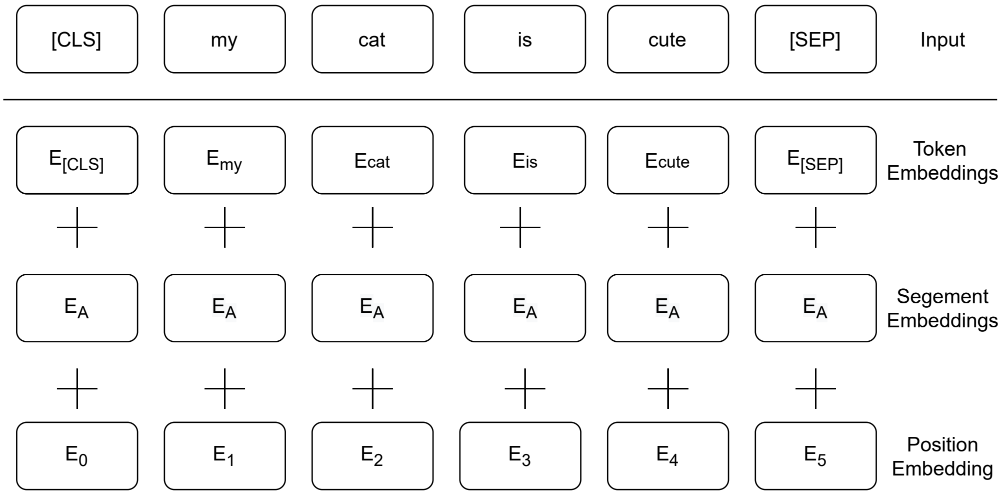


Figure 3.1: BERT input representation [8]

embedding of words. The first step of token embedding is to embed special symbols. The token [CLS] is represented as the first token of each sequence. Its output corresponding to the final transformer is used as the classification task's general sequence representation. For other tasks, this item is usually ignored. For some non-single sentence input data, there are usually two types of model processing methods. The first is to add a special [SEP] mark at the end of a single sentence to separate the single sentence. The second is sentence embedding, called Segment Embedding, which uses two markers to distinguish sentences.

As shown in Figure 3.1, subscripts A and B represent the elements of two single sentences, respectively. Unlike LSTM, Transformer does not record the location information of each word, so additional location embedding is required to record the location information of words, as shown in Figure 3.2. Location embedding will record the location information of each word in the whole sentence in turn through a unique method. For the particular case with only one sentence as input, only the embedding of sentence a needs to be used, and sentence B needs to be ignored.

Previous studies found that most deep Neural Network models adopt a unidirectional data flow model, that is, choose from left to right or from right to left according to different downstream tasks. Some recent studies combine the two directions and use the Bidirectional model to make up for the deficiency of the Unidirectional mode. Strictly speaking, this bidirectional approach is not bidirectional, so it is just a simple combination of left to right and right to left hidden layers.

Intuitively, as shown in Figure 3.2, the effect of the Deep Bidirectional model is better than the simple connection effect of the Bidirectional model. Still, the results of the real experiment may not be better. Therefore, the model provides two training tasks: the Language Masking task and the Next Sentence Prediction task.

3.1.1 Mask language model

In order to train the depth bidirectional representation ability of the model, a simple training method is adopted, the Masking Language Model (MLM) [35], as shown in Figure 3.1, this method was proposed by Taylor in 1953 [36]. Similar to the early standard language model, the final hidden vector corresponding to the masked tag will be input into the output softmax or full connection layer corresponding to the vocabulary in order to correspond the masked tag to A word in a vocabulary. Similar to the Noise reduction autoencoder [37], the model randomly shields 15% of the word tags in each sequence. The difference is that the model only predicts the shielded words rather than the sentence input of the whole model.

In the process of fine-tuning, the model does not know the masking mark of the pre-training model, resulting in a mismatch between the two. In order to optimize this situation, a new masking method is adopted in the pre-training model. First, the training data generator randomly selects 15% of the training data for labeling, then selects 80% of them for the random mask, 10% of them are selected for a random word replacement, and the remaining 10% are selected for no replacement.

In the Transformer model, we do not know which words are masked, so we do not know which words it is required to predict. Therefore, the model can only be forced to learn the context representation of each input tag. From the perspective of the whole model, only 1.5% (15%*10%) of the whole data set is replaced randomly, so from the experimental point of view, it will not greatly impact the language understanding ability of the model.

3.1.2 Next sentence prediction

In the sentiment analysis task, because the comment data text does not always appear in a single sentence, it is also accompanied by many context-related sentence texts. Therefore, adding the following sentence prediction task in the model is necessary. The training task is a simple binary classification task. There is a 50% probability that B sentences follow A and the other 50% probability that B sentences are randomly selected from the corpus in the provided pre-training example A and example B.

The choice of which sentences are actual context sentences and which do not context sentences also needs to be realized by random corpus selection. From the experimental results, the above two increased training tasks have greatly improved the accuracy of the deep bidirectional context model. In the research process of the next section, we will change some details of the model to achieve a better training effect for the comment text.

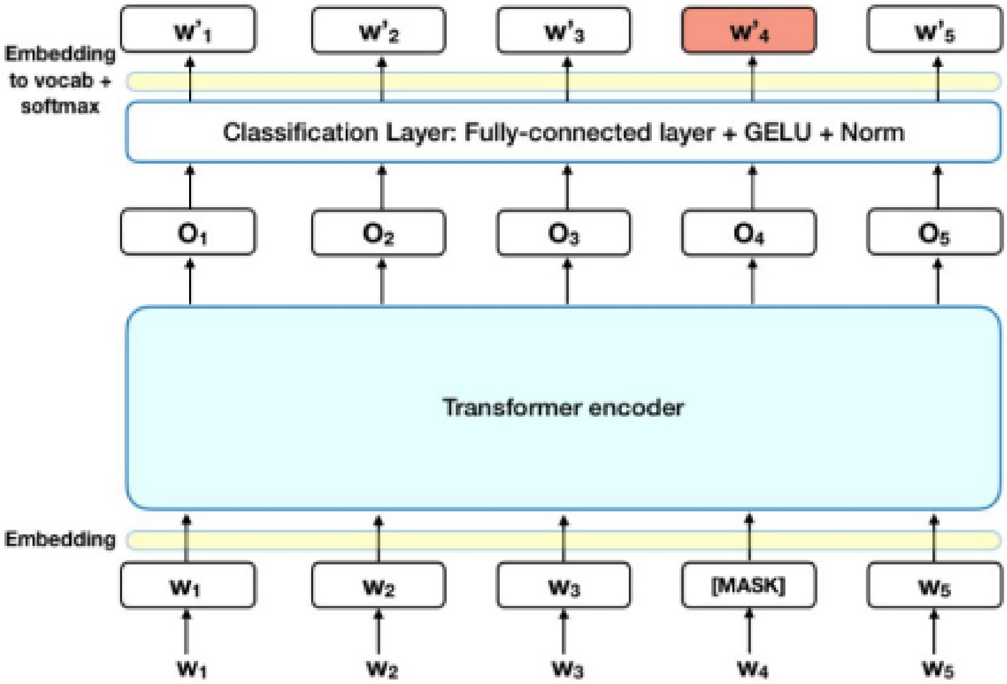

w'1
w'2
w'3
w'4
w'5
Embedding to vocab + softmax
Classification Layer: Fully-connected layer + GELU + Norm
O1
O2
O3
O4
O5
Transformer encoder
Embedding
w1
w2
w3
[MASK]
w5
w1
w2
w3
w4
w5

Figure 3.2: MLM model. [9]

3.2 The OFDP BERT model

This section will introduce the main task of this experiment in detail and present a new model of BERT, named OFDP BERT. As a general language representation model, from the perspective of the corpus, BERT has only carried out long-term pre-training on Wikipedia and books Corps. It is lacking in the field of sentiment analysis of Chinese short text.

First of all, BERT uses all English data set to train data and does not carry out a large-scale targeted training on Chinese characters, so it can not be universal in a Chinese environment. Later, Google provided a Chinese pre-training model test version, but the English Wordpiece representation is not processed in Chinese, so Wordpiece does not apply to Chinese. The test version model only modifies the English word representation Wordpiece to character-based embedding; Google only modifies the word representation, and there is no retraining of Chinese data. Therefore, BERT's performance in Chinese natural language processing tasks is not as good as in English.

In order to utilize the OFDP BERT model, this study uses Chinese short text for pre-training. We consider that BERT needs industrial hardware facilities for pre-training large-scale data. Through the experiments of multiple groups of small data sets formed by random extraction algorithms, we believe that better training results can be obtained through certain data pre-training. The experiments verify that the cost is directly proportional to the final results. The specific data information, experimental steps, and results will be given in detail in Chapter 4.

Next, the internal calculation details of the model are explained in detail. The main framework of OFDP BERT is the transformer. The core contribution of the transformer is the self-attention mechanism. Thanks to the support of self-attention technology, the whole model can take the whole sentence corpus as the model's input instead of sequence input so that the long sentences can be better represented.

The specific self-attention representation and calculation steps are shown in Figure 4.1. The input of the self-attention model is the matrix representation of sentences; Through three different full connection layers, the input matrix is transformed into three difference matrices, which are distinguished by Q (query), K (key), and V (value); through the calculation of Q and K, we can obtain the parameter weights; then, by adding these weights to the matrix V, we can obtain an available new matrix representation to complete the self-attention calculation.

Chapter 4

Experimental results

4.1 Experimental settings

This research work will be divided into two parts, experiment 1 and experiment 2. The experiment 1 includes two parts, one is the pre-training of OFDP BERT, the other is the fine-tuning process. The experiment 2 is sentiment classification experiment. The experiment 1 includes reading the comment data set document (TXT format) and data pre-processing (word segmentation). The experiment will use the word segmentation technology introduced in Chapter 2 to eliminate excessive text data. In the experiment 2, it is also necessary to pre-process the data and add a separate module for mask operation.

We run our experiment with following settings:

- Intel Core i7-5600u (3.40GHz / L3 4m) with 16GB memory
- Python 3.9 program language
- Anaconda programming environment
- TensorFlow framework
- Two rtx2080ti GPU model with 12GB memory
- Experimental data with TXT format

4.2 Data set

In the experiment of this research work, we crawled the merchant's ID, merchant name, average score, address, average price and the total number of comments. The results are saved in a TXT file, and one line represents the data of a merchant. All data sets are obtained by crawling comment data using the API extracted by Meituan open platformChina's largest online food delivery platform.

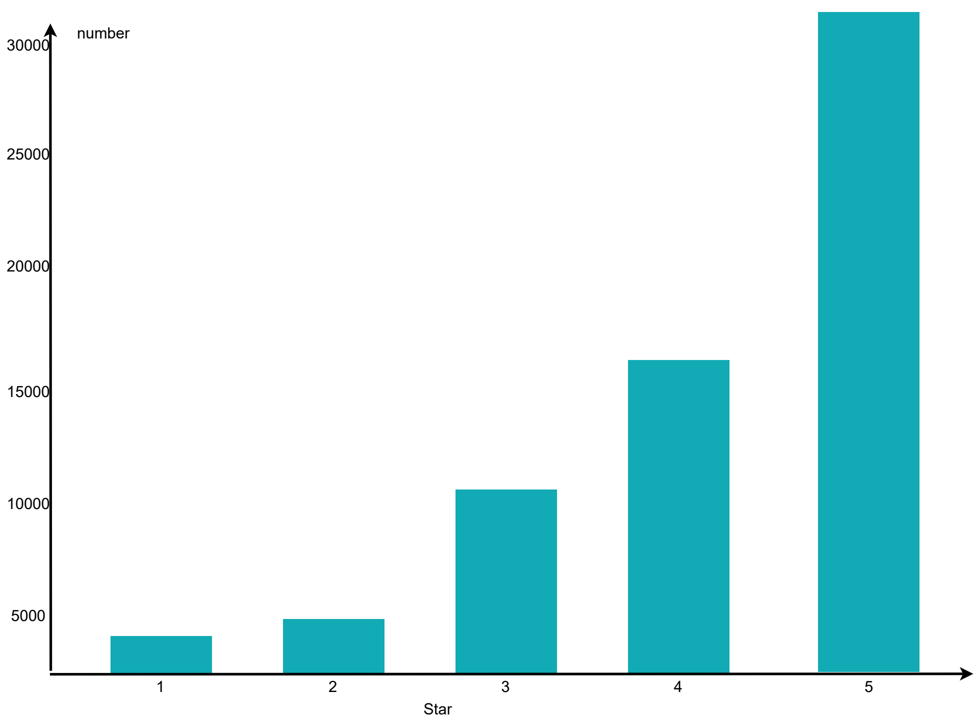


Figure 4.1: Quantity distribution of each star level.

Our experiment is only based on the text content, so we only care about the crawling comment text content. The target of data crawling is all user comments of the top 40 best-selling restaurants in Beijing in a month(June.2021). In the data pre-processing stage, the experiment carried out routine data cleaning, including discarding comments less than five words and deleting data with unrecognized symbols. Then, by sorting out a small amount of existing data on the Internet for data expansion, a total of 62243 comment texts were collected. Next, we use the word segmentation method introduced in Chapter 2 to segment words and get about 2 million words.

4.3 Experimental results and discussion

4.3.1 Experimental results

As can be seen from the table, our model performs better on the test set. The reason may be that the test set has fewer data.

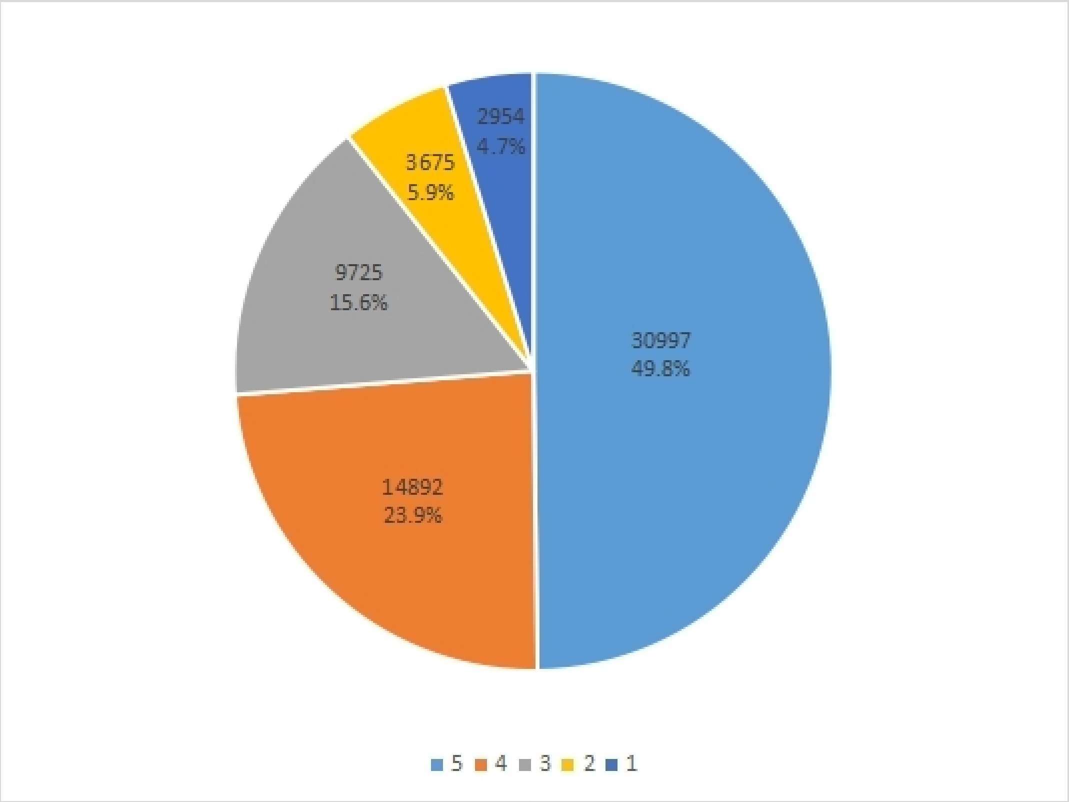


Figure 4.2: Proportion of each star level.

4.3.2 Sentiment classification

Vectorization with BERT, divide the data into input (comment) and output (star), call the encode method for vectorization, convert the multi-label classification task into multi-category classification task and save the results. Then encode the data, and classify the data according to the proportion of 8:2, that is, 80% of the positive and negative comments are selected as the training set and 20% as the test set. In order to better represent the basic situation of the data set, this paper adopts visual This method displays the training set in the data set on two attributes. Firstly, we analyze the emotional tendencies among different stars in the test set. The “0” represents

Dataset	Accuracy	Recall	F1
Train	0.94	0.92	0.93
Test	0.95	0.93	0.93

Table 4.1: Experimental results

negative emotion and “1” represents positive emotion. It can also be seen that there are a lot of negative sentiment comments in high star reviews.

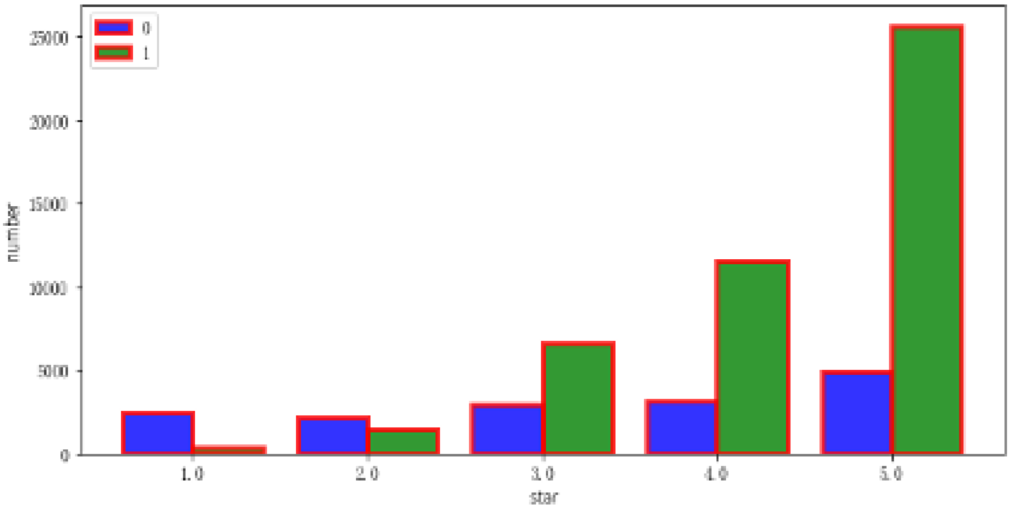


Figure 4.3: The number of labels on each fine-grained aspect.

Second is the sentiment label distribution of the training set: the histogram distribution of the number of labels of the training set in each fine-grained square is shown in the figure. We analyze which dimensions users will pay attention to, including price, dishes, location, business service, environment, and so on. We focus on the two dimensions of price and dishes. This is actually a four-category task. For each comment, we need to judge whether it is positive, negative, or neutral in terms of aspect, or not mentioned at all. This is also a single sentence category task.

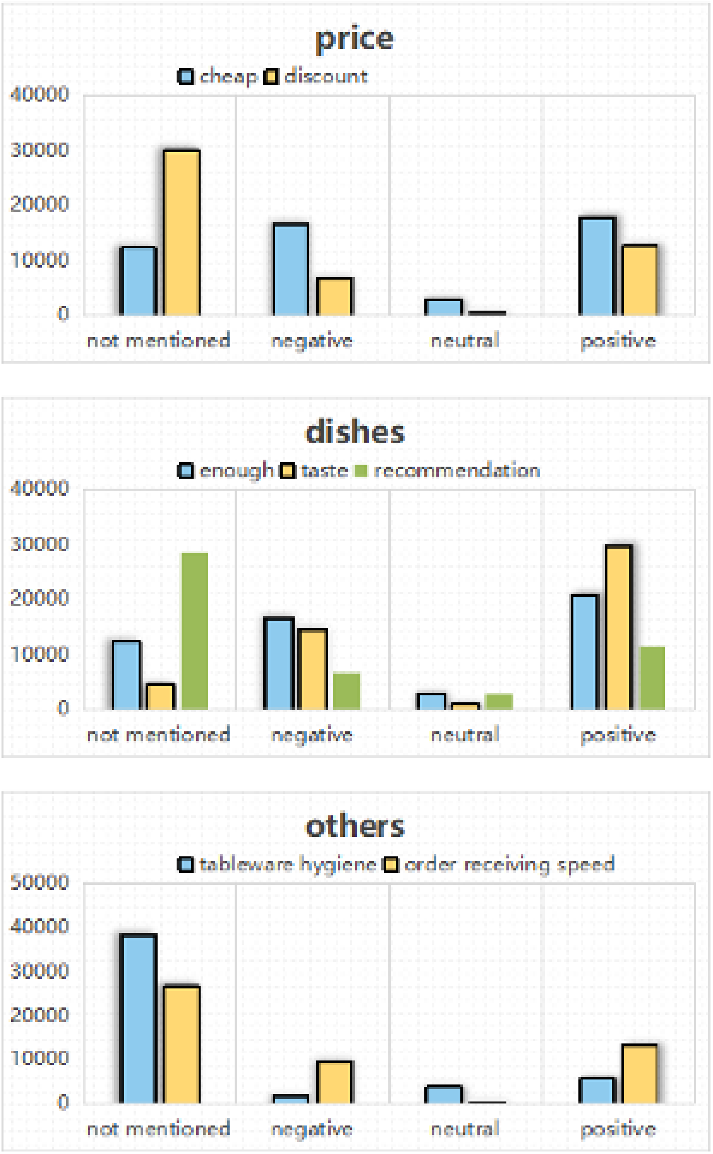


Figure 4.4: The number of labels on each fine-grained aspect.

As can be seen from Figure 4.4, the number of tags varies in different fine-grained aspects, but almost all show an unbalanced form. We can also see that the number of sentiment labels is mainly distributed in “not mentioned” and “positive”. It is impossible for consumers to mention all the fine-grained aspects in the evaluation text. At the same time, consumers also tend to evaluate the parts they feel very satisfied with. Therefore, there are more “not mentioned” and “positive” sentiments in the overall distribution, which is more appropriate to the actual distribution.

4.3.3 Evaluation

In this part, we show the visualization results under each fine-grained aspect. This experiment counted three kinds of sentiment attitudes in seven fine-grained aspects: good rate, mid rate, and bad rate. The calculation of good rate is showed as follows positive num is the number of sentiment labels that are “positive” and negative num is the number of sentiment labels that are “negative”, neural num is the number of sentiment labels that are “neutral”. To prevent a denominator of 0, add +1 to the denominator. The other two calculation formulas of sentiment attitude are the same.

The following figure shows the statistical display of the histogram of the distribution of sentiment tendencies in the two fine-grained aspects most concerned by customers. The higher the good rate, the more “positive” sentiment tendencies of the consumer group under the merchant.

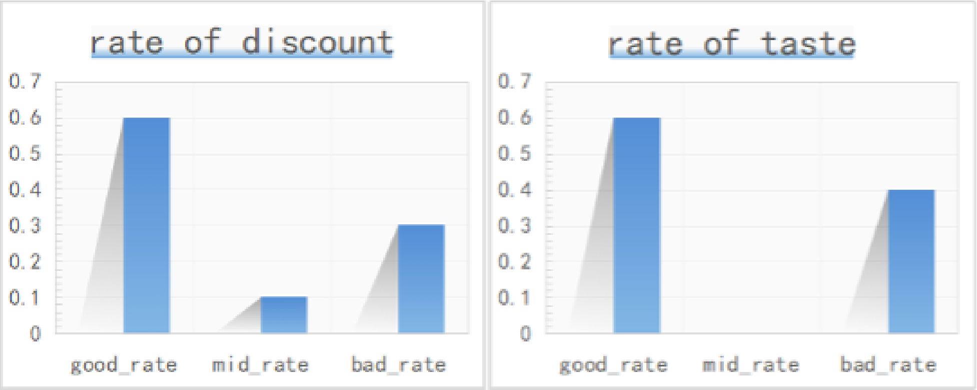


Figure 4.5: Sentiment tendency statistics under two fine-grained aspects.

According to the above good rate ratio, we randomly selected the sentiment of several merchants’ comments and the hot-ranking data of their stores for linear regression analysis.

4.3.4 Word-cloud on stars level

From the results of regression analysis, there is a strong correlation between ranking and good rate. We can conclude that the positively of comments is directly proportional to the popularity ranking of merchants. Finally, we made five Word cloud graphs according to the stars of the comment data set.

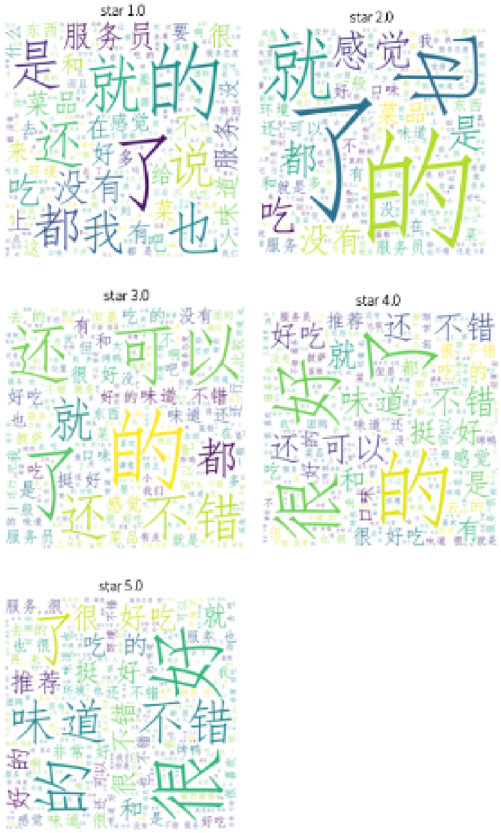


Figure 4.6: Word cloud on stars.

As shown in the figure, the higher the star level, the greater the proportion of “很”(Very) and “好”（Good）in the Word cloud, and the higher the degree of positive sentiment of customers.

Chapter 5

Conclusion

5.1 Summary

This thesis studies the BERT model with significant influence in the NLP field and its application for the online food delivery platforms. In detail, we reset the whole pre-training process to train OFDP BERT model by using the Meituan's Chinese comment data set. Then, we fine-tune the parameters of the last layer of the pre-trained model.

We further add an attention layer to the output of the coding layer of the original BERT to weight with different positions and finally take the weighted features as classification features. Text and aspect phrases are embedded into the model at sentence level for fine-tuning procedure. The multi-label classification task is transformed into a multi-category classification task. Finally, sentiment analysis performance is analyzed using the above trained model, which proves the positive significance of this study.

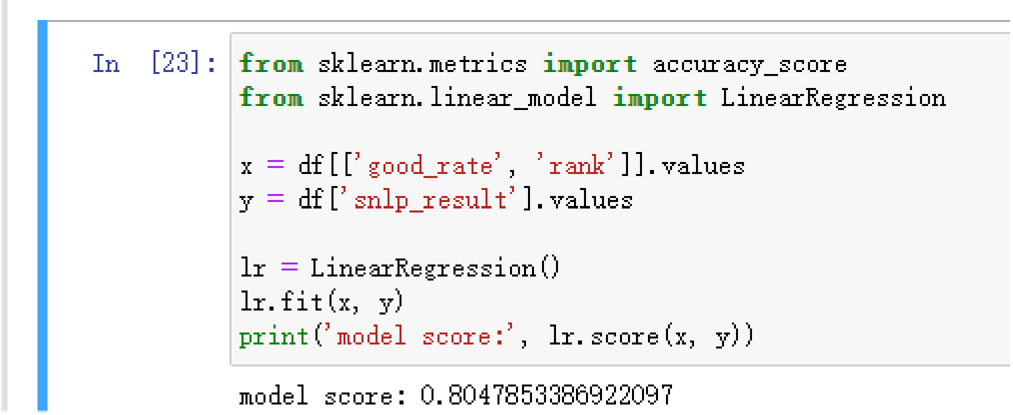
5.2 Limitations and future directions

According to the experimental results, the rising space of the model and the research space of the problem are still considerable. Firstly, in the part of pre-training, we can increase the number of corpora in this field in the future to make a more accurate judgment in the first step of emotion recognition. The training method used in this paper is relatively simple, so the room for improvement is rather large. In addition, we have not considered the issue of long text at present, and we may have to improve the comprehensive support of long text in future work. Nowadays, many Internets communication tools support long text editing, which shows that the social media environment of long text is an inevitable problem.

In addition, it is not just the problem of BERT model. It may exist in many deep learning models. The classification results or prediction results change when doing some simple word replacement, which is a research direction. How to improve the robustness and stability of the current pretraining language model.

Appendix A

Code Example



```
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LinearRegression

x = df[['good_rate', 'rank']].values
y = df['snlp_result'].values

lr = LinearRegression()
lr.fit(x, y)
print('model score:', lr.score(x, y))
```

```
model score: 0.8047853386922097
```

Figure A.1: Linear regression display.

References

[1] W. Junfei. Recurrent neural network(1):architecture. [Online]. Available: https://www.cnblogs.com/rhyswang/p/9091534.html

[2] Colah. Understanding lstm networks. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[3] S. Ballı and O. Karasoy, "Development of content-based sms classification application by using word2vec-based feature extraction," *IET Software*, vol. 13, no. 4, pp. 295–304, 2019.

[4] R. SEBASTIAN. Deep learning for nlp best practices. [Online]. Available: https://ruder.io/deep-learning-nlp-best-practices/

[5] Unname. Word segmentation algorithm based on forward maximum matching algorithm. [Online]. Available: https://www.programmersought.com/article/35136302697/

[6] S. Biswas and J. Gall, "Structural recurrent neural network (srnn) for group activity analysis," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 1625–1632.

[7] B. Jang, M. Kim, I. Kim, and J. W. Kim, "Eagleeye: A worldwide disease-related topic extraction system using a deep learning based ranking algorithm and internet-sourced data," *Sensors*, vol. 21, no. 14, p. 4665, 2021.

[8] S. Paul and S. Saha, "Cyberbert: Bert for cyberbullying identification," *Multimedia Systems*, pp. 1–8, 2020.

[9] U. Sharma, P. Pandey, and S. Kumar, "A transformer-based model for evaluation of information relevance in online social-media: A case study of covid-19 media posts," *New Generation Computing*, pp. 1–24, 2022.

[10] J. Kim, J. Kim, and Y. Wang, "Uncertainty risks and strategic reaction of restaurant firms amid covid-19: Evidence from china," *International Journal of Hospitality Management*, vol. 92, p. 102752, 2021.

[11] I. Tenney, D. Das, and E. Pavlick, "Bert rediscovers the classical nlp pipeline," *arXiv preprint arXiv:1905.05950*, 2019.

[12] R. Socher, Y. Bengio, and C. D. Manning, "Deep learning for nlp (without magic)," in *Tutorial Abstracts of ACL 2012*, 2012, pp. 5–5.

[13] Q. Jiang, L. Chen, R. Xu, X. Ao, and M. Yang, "A challenge dataset and effective models for aspect-based sentiment analysis," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 6280–6285.

[14] Y. Song, J. Wang, Z. Liang, Z. Liu, and T. Jiang, "Utilizing bert intermediate layers for aspect based sentiment analysis and natural language inference," *arXiv preprint arXiv:2002.04815*, 2020.

[15] R. Feldman, "Techniques and applications for sentiment analysis," *Communications of the ACM*, vol. 56, no. 4, pp. 82–89, 2013.

[16] B. Liu, "Sentiment analysis and opinion mining," *Synthesis lectures on human language technologies*, vol. 5, no. 1, pp. 1–167, 2012.

[17] H. Xu, B. Liu, L. Shu, and P. S. Yu, "Bert post-training for review reading comprehension and aspect-based sentiment analysis," *arXiv preprint arXiv:1904.02232*, 2019.

[18] Y. Ma, H. Peng, and E. Cambria, "Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[19] J. Schmidhuber, "Learning complex, extended sequences using the principle of history compression," *Neural Computation*, vol. 4, no. 2, pp. 234–242, 1992.

[20] Y. LeCun, D. Touresky, G. Hinton, and T. Sejnowski, "A theoretical framework for back-propagation," in *Proceedings of the 1988 connectionist models summer school*, vol. 1, 1988, pp. 21–28.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[23] Z. Yun-tao, G. Ling, and W. Yong-cheng, "An improved tf-idf approach for text classification," *Journal of Zhejiang University-Science A*, vol. 6, no. 1, pp. 49–55, 2005.

[24] E. Altszyler, M. Sigman, S. Ribeiro, and D. F. Slezak, “Comparative study of lsa vs word2vec embeddings in small corpora: a case study in dreams database,” *arXiv preprint arXiv:1610.01520*, 2016.

[25] H. Jelodar, Y. Wang, C. Yuan, X. Feng, X. Jiang, Y. Li, and L. Zhao, “Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey,” *Multimedia Tools and Applications*, vol. 78, no. 11, pp. 15 169–15 211, 2019.

[26] R. Mehrotra, S. Sanner, W. Buntine, and L. Xie, “Improving lda topic models for microblogs via tweet pooling and automatic labeling,” in *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, 2013, pp. 889–892.

[27] B. Dong and X. Wang, “Comparison deep learning method to traditional methods using for network intrusion detection,” in *2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)*. IEEE, 2016, pp. 581–585.

[28] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, “A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains,” *The annals of mathematical statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[29] L. Kong, C. Dyer, and N. A. Smith, “Segmental recurrent neural networks,” *arXiv preprint arXiv:1511.06018*, 2015.

[30] G. Andrew, “A hybrid markov/semi-markov conditional random field for sequence segmentation,” in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, 2006, pp. 465–472.

[31] J. Zhuo, Y. Cao, J. Zhu, B. Zhang, and Z. Nie, “Segment-level sequence modeling using gated recursive semi-markov conditional random fields,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 1413–1423.

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

[33] B. McCann, J. Bradbury, C. Xiong, and R. Socher, “Learned in translation: Contextualized word vectors,” *arXiv preprint arXiv:1708.00107*, 2017.

[34] Z. Huang, D. Liang, P. Xu, and B. Xiang, "Improve transformer models with better relative position embeddings," *arXiv preprint arXiv:2009.13658*, 2020.

[35] A. Conneau and G. Lample, "Cross-lingual language model pretraining," *Advances in Neural Information Processing Systems*, vol. 32, pp. 7059–7069, 2019.

[36] W. L. Taylor, ""cloze procedure": A new tool for measuring readability," *Journalism quarterly*, vol. 30, no. 4, pp. 415–433, 1953.

[37] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 1096–1103.