修 士 論 文 概 要 書

Master's Thesis Summary

Date of submission: 01/25/2021 (MM/DD/YYYY)

専攻名 (専門分野) Department	Computer Science and Communications Engineering	氏 名 Name	ZHOU YUCHENG	指導教員 Advisor	渡辺 裕 印 Seal
研究指導名 Research guidance	Research on Audiovisual Information Processing	学籍番号 Student ID number	CD 5119F044-04		
研究題目 Title	Attention-Pooling Text Classification Model based on Pre-training Language Model				

1. Introduction

Today, the information revolution represented by big data and cloud computing is in the ascendant, and information technology has penetrated into all aspects of social development. The rapid development of Internet technology has greatly enriched information and data resources. These massive data have various formats, including text, voice, images, etc. And all these data are still growing explosively. According to statistics, the amount of data generated on the Internet each year is more than the sum of all data generated in the earlier history of mankind, and more than 80% of those data was text data. Obtaining useful information from these text data has become a concern of people.

The Internet contains so much text data that it is impossible to simply browse and analyze the information one by one. Therefore, Information technologies such as text mining and information retrieval have emerged. The type of text mining include classification, clustering, summary, trend prediction, etc. In order to adapt to the characteristics of Internet text, all these applications require text processing technology to be able to process a large amount of unstructured data. Especially, information retrieval needs to manipulate a large amount of data in a short time, so If we can provide a good organization and structure for the text, we will save a lot of time and cost. The automatic text classification technology can efficiently organize similar and related texts together, providing strong support for the upper-level tasks of data mining and the efficient query of information retrieval.

2. Related Work

Text classification means that the computer divides the text into pre-defined categories according to the text content by using an automatic classification algorithm. Text classification is one of the basic tasks in the field of NLP. Generally, a text classification model can be divided into three steps: vector representation of text, feature extraction of text, and construction of classifiers. According to the number of labels, text classification tasks can be divided into binary classification problems and multiple classification problems.

The history of text classification can be traced back to the beginning of machine learning algorithms in the middle of the twentieth century, when computers were mainly used to process small-scale structured text data. Various machine learning algorithms, such as Support Vector Machine (SVM) [1], Naive Bayes classifiers, and Maximum Entropy Model, have been well applied in text classification tasks. Because text data is close to the natural language that people used in daily, it is particularly concerned by researchers. Therefore, many NLP related fields have been developed, such as sentiment analysis, relationship extraction and spam detection, etc. Text classification is an important part of all these NLP research directions.

The advent of the big data era is also accompanied by the establishment of large-scale data sets. The performance of machine learning methods on large-scale data has begun to be inadequate, so deep learning methods represented by deep neural networks have gradually become popular in the field of NLP and applied in classification tasks. There are some common models such as Recurrent Neural Networks (RNN) [2] and Convolutional Neural Networks (CNN) [3]. Deep learning methods can extract more abstract features of text, and perform well in the case of a large amount of data. Compared with machine learning algorithms, they have relatively lower requirements of quality about input data, and are more suitable for processing large-scale data. The application of deep learning methods has greatly improved the accuracy of text classification, thereby enhancing the development of various downstream classification tasks such as sentiment analysis, and also promoting the development of industrial applications such as reading comprehension, search engines, and public opinion monitoring. Until today, text classification is still a very active research direction in the field of NLP.

As a formal research field, text classification began roughly in the 1960s. At that time, the main method was to use Knowledge Engineering. The method of it was to use the rules of manually labeling data, which required people who has a good understanding of linguistics, so most of the early NLP research has the participation and help of linguists. However, this method is very labor intensive and inefficient. After the proposal of machine learning algorithm, they were only used as some auxiliary means on text classification tasks. From the 1990s to the large-scale application of deep neural networks, machine learning algorithms performed well in text classification. In recent years, the attention model [4] has successfully replaced part of the structure

of the neural network model in some complex tasks. Since 2018, deep learning methods based on pre-training models have achieved breakthrough results in various NLP tasks. Especially in the text classification field, many experiments have achieved better results than before, but also require much higher training cost at the same time.

3. Methodology

Aiming at the shortcomings of the current standard deep learning algorithm training rate and the shortcomings of the accuracy of Chinese multi-label text classification, we propose a text classification method based on pre-training language model, attention mechanism and pooling mechanism. The attention part of the model can focus on the more important local features of the text data itself, and the pooling part can obtain the overall features by directly operating the word vector.

As shown in Figure 1, The model mainly consists of three parts: word embedding layer, pooling layer and attention layer.

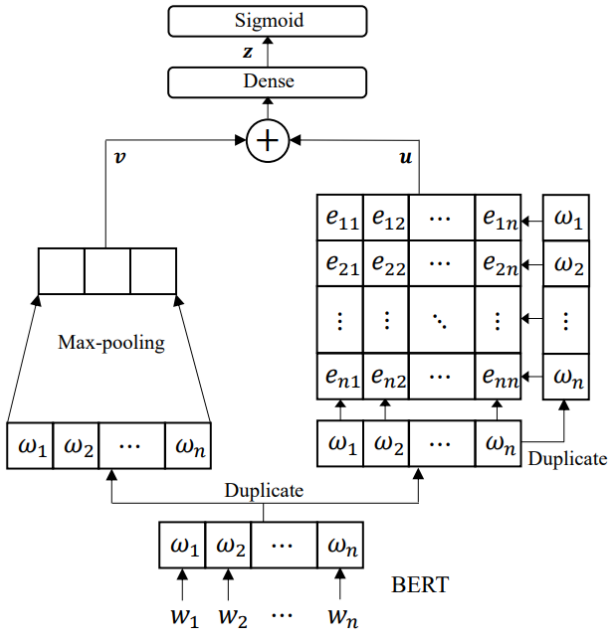

Figure 1: Structure of the proposed model

4. Data

We have established a small-scale Chinese multi-label text data set, and the source of the corpus is the Chinese Internet forum. The reason for choosing this type of text is that there are many and complex speeches in the forum, and valuable information is often mixed with a lot of meaningless information. Even with the current search engine technology, it is difficult to filter out high-value information. This characteristic is very suitable for verifying model performance.

This data set contains 5000 texts of various lengths and 10 labels. We use manual labeling to give relevance to text and labels. At the same time, we have also performed batch preprocessing on all texts, such as replacing hyperlinks in the text with ordinary text, deleting emoji characters in the text, etc.

5. Experiment

In order to verify the effectiveness of the proposed model, this paper compares it with CNN, LSTM, and APCNN. Among them, TextCNN and LSTM are used as a comparison of standard deep learning models. The evaluation methods of this paper are Macro Precision Rate (MacroP), Macro Recall Rate (MacroR), and Macro F1 value (MacroF1).

6. Result and Analysis

The comparison results of the accuracy of each model on the test set are shown in Table 1.

Table 1: Model comparison results

	Acc/%	P/%	R/%	F1/%
TextCNN	73.74	74.09	78.03	76.01
LSTM	71.30	70.33	71.98	71.15
APCNN	73.78	73.00	71.12	72.05
Ours	83.29	77.51	75.28	76.38

It can be seen from the table that the model proposed in this paper has an accuracy rate of 83.29% on the test set, which is higher than the neural network model TextCNN, LSTM and the attention model APCNN. In terms of macro precision, the proposed model is also superior to all other models in the experiment. But in terms of recall rate, it is slightly lower than CNN, because the CNN model also has a pooling layer mechanism, which is suitable for classification tasks. It shows that this model is not particularly prominent in coverage. Also see that the accuracy of CNN is 73.74%, which is lower than the two attention models. This may be because although CNN can extract local features and then synthesize them into global features, it is difficult to pay attention to the important features of the overall text. The higher MacroF1 value of the proposed model indicates that it performs well overall.

7. Conclusion

This paper proposes a model based on attention and pooling mechanism. In order to shorten the convergence time, this model does not use a complex deep neural network structure in feature extraction step. Instead, it uses attention to extract relatively important local features of the text, uses pooling to extract the global features of the text, and combines those advantages to apply into text classification tasks.

Reference

[1] Tong S, Koller D. Support vector machine active learning with applications to text classification[J]. The Journal of Machine Learning Research, 2002, 2(1):999-1006.

[2] Jordan M I, Ghahramani Z, Jaakkola T S, et al. An introduction to variational methods for graphical models [J]. Machine learning, 1999, 37(2): 183-233.

[3] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition [J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[4] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Proceedings of the 31st Annual Conference on Neural Information Processing Systems. California: NIPS,2017:5998-6008.

Attention-Pooling Text Classification Model based on Pre-training Language Model

A Thesis Submitted to the Department of Computer Science and Communications Engineering,
the Graduate School of Fundamental Science and Engineering of Waseda University
in Partial Fulfillment of the Requirements for the Degree of Master of Engineering

Submission Date: January 25th, 2021

Yucheng ZHOU
(5119F044-4)

Advisor: Prof. Hiroshi Watanabe
Research guidance: Research on Audiovisual Information Processing

Acknowledgements

To begin with, I would like to express my sincerest gratitude to my research supervisor Prof. Hiroshi Watanabe, who give me the most enthusiastic guidance and sparing no effort support. Without his guidance and support, it will be impossible for me to finish my study and research in past two years.

Then, I should express my heartfelt gratitude to every member in Watanabe Lab, who have accompanied me through the 2 years at Waseda University, sharing happiness and sadness with me.

Finally, my thanks would go to my beloved family for their constant overall support and loving considerations through my all life.

Abstract

Text classification refers to automatically dividing text data into predetermined categories by computers. It is a basic task in natural language processing (NLP) and plays an important role in the fields of information retrieval and data mining. Based on the deep learning classification method and pre-training language model, this paper proposed an attention-pooling text classification model that combined the attention mechanism and pooling mechanism to extract the features from the text data.

In order to train and verify the model we proposed, we built a small-scale Chinese multi-label data set. The corpus comes from Internet forums and has been divided into 10 categories.

By further evaluating the performance of the proposed method, we trained several models with our data set, and compare the accuracy and other evaluation method outputting by test-set. As a result, wo confirmed that the proposed model has a good performance on text classification tasks and also has a good convergence speed.

Key words: Text classification, Attention mechanism, Pooling, Natural Language processing

List of Contents

Chapter 1 Introduction

1.1 Background

Today, the information revolution represented by big data and cloud computing is in the ascendant, and information technology has penetrated into all aspects of social development. The rapid development of Internet technology has greatly enriched information and data resources. These massive data have various formats, including text, voice, images, etc. And all these data are still growing explosively. According to statistics, the amount of data generated on the Internet each year is more than the sum of all data generated in the earlier history of mankind, and more than 80% of those data was text data. Obtaining useful information from these text data has become a concern of people.

The Internet contains so much text data that it is impossible to simply browse and analyze the information one by one. Therefore, Information technologies such as text mining and information retrieval have emerged. The type of text mining include classification, clustering, summary, trend prediction, etc. In order to adapt to the characteristics of Internet text, all these applications require text processing technology to be able to process a large amount of unstructured data. Especially, information retrieval needs to manipulate a large amount of data in a short time, so If we can provide a good organization and structure for the text, we will save a lot of time and cost. The automatic text classification technology can efficiently organize similar and related texts together, providing strong support for the upper-level tasks of data mining and the efficient query of information retrieval.

1.2 Problem Statement

Natural Language Processing (NLP) is an important research direction in the field of computer intelligent information processing. It refers to the use of computers to perform

morphological, syntactic and grammatical analysis operations to achieve the purpose of identifying, understanding and analyzing human language, thereby completing various follow-up tasks, such as named entity recognition, relationship extraction, text classification, automatic summarization, question answering system, sentiment analysis, machine translation, etc. NLP faces many obstacles under traditional machine learning methods. In the other side, in recent years, deep learning-based models have brought new and broad prospects to NLP and achieved remarkable results in various practical tasks. NLP has a wide range of applications: personalized recommendation can learn the user's personal preferences based on the user's historical behavior records, thereby predicting the user's preference for a given product, and achieving an accurate understanding of the user's intention; Question answering system can correctly analyze the question posed by the user in natural language, extract the important information in the question, automatically find the accurate answer from various data resources, and return the searched answer to the user; sentiment analysis can automatically analyze the tone, emotion and credibility in the text, as well as making judgments on the quality of public opinion, help companies analyze customer consumption trends, understand policies, analyze hot topics, and timely monitor crisis public opinion.

Text classification can be divided into single-label classification and multi-label classification according to the number of labels to be predicted. Traditional single-label classification refers to a classification problem in which each sample can only correspond to one label. This situation is not common in practical applications. With the diversified development of information and the rapid increase in the amount of information, an instance may be associated with multiple labels. For example, a news may belong to both "technology" and "astronomy.", as shown in Figure 1.1. Multi-label classification tasks refer to the problem that a sample may belong to multiple labels at the same time.

> The heavy-lift Long March 5 lifted off from the Wenchang Satellite Launch Center at 3:30 p.m. Eastern. The Chang'e-5 spacecraft was announced to have successfully entered its predetermined orbit around 4:45 p.m., following deployment of solar arrays. The 8.2-ton Chang'e-5 spacecraft is to begin an estimated 112-hour journey to the moon.

Figure 1.1 A piece of news about the takeoff of the Chang'e-5

When dealing with large-scale unstructured data, deep learning algorithms are more effective. Compared with statistical machine learning methods, deep learning has the advantages of high accuracy and strong flexibility. But it also has certain shortcomings, such as massive parameters and high time complexity, which lead to a large time cost of

model training and required huge computing resources. In response to these shortcomings, some researchers have proposed neural network models with a more simplified structure to speed up the iteration speed on the basis of retaining the good feature extraction capabilities of deep learning algorithms. The model proposed in this paper is improved on the basis of traditional deep neural networks to improve the performance of the model on text classification tasks and better serve the upper-level NLP tasks.

1.3 Outline

The outline of this thesis is organized as follows:
Chapter 1: We describe the background of NLP and text classification and introduce the application scenarios and value of these technologies. Besides, we state the current problems facing the text classification field and mention some possible solutions

Chapter 2: Starting from the text representation method, we introduced the development history of various text classification methods. After describing the advantages and limitations of various traditional text classification methods, two classic deep learning methods are described in detail. Finally, we share the latest research progress in recent years.

Chapter 3: We explained the text classification method we proposed, introduced the principle and application method of the attention mechanism, and explained the structure and function of the main network layer

Chapter 4: In order to train and verify our proposed model, we introduced a small-scale Chinese multi-label dataset that we built. We also use this data set to train and compare various models, and analyze their advantages and disadvantages.

Chapter 5: Concludes this thesis.

Chapter 2 Previous Work

In recent years, text classification research has developed rapidly, and specific applications have also increased. This is inseparable from the proposal of distributed vector representation and the continuous improvement of classification algorithms. From early knowledge engineering and statistical machine learning to deep learning algorithms, to Attention model and the most recent Pre-training Language model, the performance of classification models in each period has made a qualitative leap.

2.1 Text Classification

Text classification means that the computer divides the text into pre-defined categories according to the text content by using an automatic classification algorithm. Text classification is one of the basic tasks in the field of NLP. Generally, a text classification model can be divided into three steps: vector representation of text, feature extraction of text, and construction of classifiers. According to the number of labels, text classification tasks can be divided into binary classification problems and multiple classification problems.

The history of text classification can be traced back to the beginning of machine learning algorithms in the middle of the twentieth century, when computers were mainly used to process small-scale structured text data. Various machine learning algorithms, such as Support Vector Machine (SVM), Naive Bayes classifiers, and Maximum Entropy Model, have been well applied in text classification tasks. Because text data is close to the natural language that people used in daily, it is particularly concerned by researchers. Therefore, many NLP related fields have been developed, such as sentiment analysis, relationship extraction and spam detection, etc. Text classification is an important part of all these NLP research directions.

The advent of the big data era is also accompanied by the establishment of large-scale data sets. The performance of machine learning methods on large-scale data has begun to be inadequate, so deep learning methods represented by deep neural networks have gradually become popular in the field of NLP and applied in classification tasks. There

are some common models such as Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN). Deep learning methods can extract more abstract features of text, and perform well in the case of a large amount of data. Compared with machine learning algorithms, they have relatively lower requirements of quality about input data, and are more suitable for processing large-scale data. The application of deep learning methods has greatly improved the accuracy of text classification, thereby enhancing the development of various downstream classification tasks such as sentiment analysis, and also promoting the development of industrial applications such as reading comprehension, search engines, and public opinion monitoring. Until today, text classification is still a very active research direction in the field of NLP.

As a formal research field, text classification began roughly in the 1960s. At that time, the main method was to use Knowledge Engineering. The method of it was to use the rules of manually labeling data, which required people who has a good understanding of linguistics, so most of the early NLP research has the participation and help of linguists. However, this method is very labor intensive and inefficient. After the proposal of machine learning algorithm, they were only used as some auxiliary means on text classification tasks. From the 1990s to the large-scale application of deep neural networks, machine learning algorithms performed well in text classification. In recent years, the attention model has successfully replaced part of the structure of the neural network model in some complex tasks. Since 2018, deep learning methods based on pre-training models have achieved breakthrough results in various NLP tasks. Especially in the text classification field, many experiments have achieved better results than before, but also require much higher training cost at the same time.

2.2 Text Representation

Text representation is the first step of NLP, through this procedure, we can represent text into a vector form that can be recognized and processed by a computer. No matter which language or writing style, words is the basic unit of text. Therefore, the vectorized representation of general text is the representation of words, and the vectorized representation of words is the basis of text classification. Its quality directly affects the result of feature extraction, which determines the accuracy of text processing result. The vectorized representation of text has greatly promoted the development of text

classification because of its ease of processing large-scale data and adapting to deep learning algorithms. The representation of words includes one-hot representation, distributed representation based on statistics, and distributed representation based on neural network. In the vectorized representation of words, usually each dimension of the vector represents one part of the information about the words, such as part of speech, word order, word frequency, etc. Therefore, text representation plays an important role in text feature extraction.

2.2.1 One-hot Representation

One-hot representation means that all the words contained in the text will be assigned a unique vector. If the number of all words in the corpus is l, the vector dimension of One-hot representation is also l. In detail, first we count all the words in the corpus to obtain a vocabulary table, and then build an index for each word. Then they will be assigned to a vector with the same length of the vocabulary table, where the position corresponding to the index of that word will be assigned a *value* 1, and the other positions will be assigned a *value* 0, as shown in Figure 2.1. Obviously, each vector can only have one *value* 1. For example, if the One-hot vector represented for "likes" is [0, 1, 0, 0, 0, 0], it means that there are 6 words in the dictionary, and "likes" is the third words.

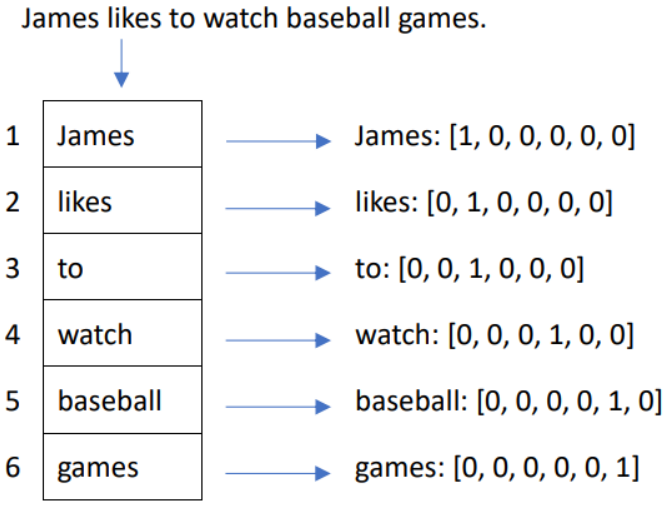

Figure 2.1 A sample of One-hot Representation

2.2.2 Bag of Words

Bag of words (BOW) representation, also known as Count Vectors representation, is to represent a text as the bag of its words, disregarding word order and relation but keeping multiplicity. Therefore, the BOW representation vector cannot show the correlation between words. For example, it is impossible for computer to understand the similarity of words "learning" and "studying". In addition, because the BOW model uses a sparse matrix to store the words, when the scale of corpus is huge, the vocabulary table will be too large to cause dimensional disasters, which is not conducive to the calculation and processing of the integral function. In short, the BOW model is only used in the early stages of NLP, and is gradually eliminated after the advent of Distributed Vector Representation. The problems mentioned above have also been gradually solved with the emergence of Distributed Vector Representation.

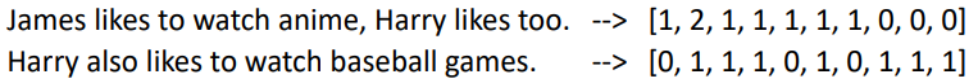

Figure 2.2 vectors of BOW Representation

2.2.3 Distributed Representation

Different from One-hot, which is a discrete representation, distributed representation is a continuous representation. It includes two types: distributed representation based on statistics and distributed representation based on neural network models.

Distributed representation based on statistics (count-based distributed representation) refers to the word frequency counted from the entire document. Each word corresponds to a different word frequency, and then assigns each word to a high-dimensional vector related to the word frequency, which is equivalent to mapping the word frequency to a higher-dimensional space. The advantage of this method is that it combines the word frequency information in the corpus and reflects the relationship between words, but it still fails to reflect the deep internal information of semantics, part of speech, word order and so on.

The concept of distributed representation based on neural network was proposed by Mikolov et al. [23]. The main methods are named as Word2Vec and Glove. Both two methods are based on neural network algorithms. Word2vec can be divided into two

types: Continuous Bag of Words (CBOW) and Skip-gram. The CBOW model is shown on the left side of Figure 2.3.

The CBOW model judges a word from its context, similar to the cloze problem, worked as predicting a specific word. Skip-Gram is shown on the right side of Figure 2.3. The principle of Skip-Gram and CBOW is opposite. Given a word, it will predict the probability distribution of each word in the context.

Glove means a global word vector, which is a word vector v_i that combines global word features. Firstly, given a pair of words and calculate the probability error square between them, then use this probability difference as the loss function of the neural network for training, and finally take the output of the neural network as the word vector. This method can represent the deep correlation between words, and can understand synonyms and antonyms.

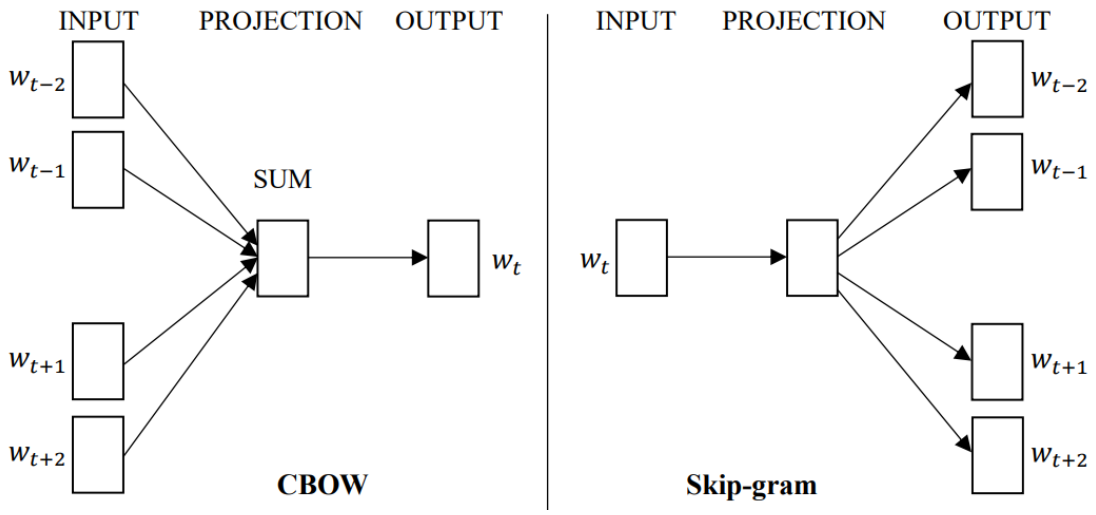

Figure 2.3 The overview of Word2Vec

2.3 Statistical Machine Learning

Since the 1990s, before the concept of deep learning was proposed, statistical machine learning was the main method in the field of NLP. The main classification algorithms include SVM, K-nearest neighbor algorithm (KNN), Naive Bayes, etc. [25]. This type of algorithm replaces the earlier classification model based on knowledge engineering and rule statistics, pays more attention to the automatic mining of text by the model itself, and improves the classification ability of labels through the continuous optimization of the algorithm itself, so it is more flexible and accurate. Therefore, machine learning

algorithms have become a common method of text classification fields.

In general, when using the traditional machine learning algorithm to construct the classification model, the first thing we need to do is manually designing the text features. Then train the model to learn the features, and finally get the classification result. These models had obtained excellent results on small-scale data sets, and were fast in calculation. However, the disadvantages of such methods are that they need to manually label text features in advance, require high quality of the data set, and rarely consider semantic and word meaning information.

2.3.1 Support Vector Machine

In machine learning, SVM is a classic representative of the kernel method, which was proposed by Cortes et al. in 1995 [26]. SVM is a classic binary classification algorithm, belongs to supervised learning algorithm. The goal of the algorithm is to find a segmentation hyperplane so that the two types of data can be separated correctly as much as possible. Although the convergence speed of SVM on large-scale dataset is slow, and requires relatively large storage resources and high computing power, its linear classification mode effectively overcomes the influence of the redundant features of sample distribution and over-fitting, has a good generalization ability [27].

The specific algorithm of SVM is as follows: Given a binary classification data set $D = \{(x, y)\}$, and $y \in \{+1, -1\}$, if the two types of samples are linearly separable, that means there is a hyperplane that can divide them into two types, as in the formula (2.1).

$$\omega^T x + b = 0 \tag{2.1}$$

where ω is the model parameter and b is the intercept.

The distance γ from each sample x to the hyperplane in the data set D is shown in formula (2.2).

$$\gamma = \frac{\|\omega^T x + b\|}{\|\omega\|} \tag{2.2}$$

In the data set, all sample points satisfying $\omega^T x + b = \pm 1$ are called support vectors. In a data set that can be linearly separated, there are many segmentation hyperplanes, but there is only one plane that can maximize the support vector separation, which is the search target of SVM. As shown in Figure 2.4, the sample points on the straight line $\omega^T x + b = \pm 1$ are support vectors.

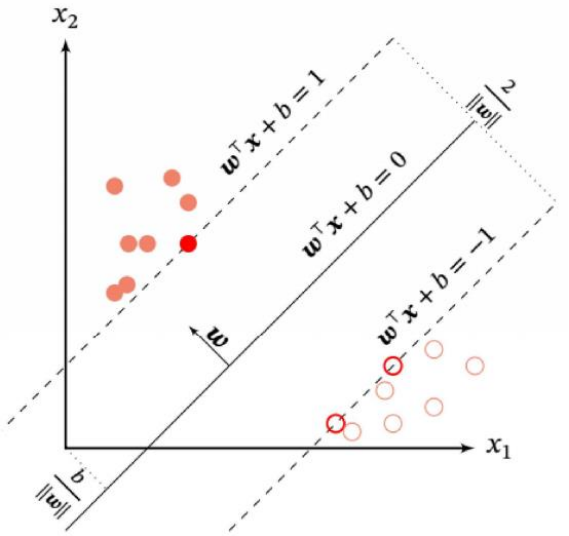

Figure 2.3 The overview of SVM

2.3.2 K-Nearest Neighbor

K-Nearest Neighbor (KNN) was proposed by Cover et al. [28], which is an example-based learning method. The KNN algorithm assumes that each sample corresponds to a point in an n-dimensional space, and the nearest neighbor point of each sample is defined by Euclidean distance, so the sample can be expressed as a feature vector, as shown in formula (2.3).

$$\langle a_1(x), a_2(x), \dots, a_n(x) \rangle \tag{2.3}$$

In the formula, $a_n(x)$ represents the nth attribute value in the instance x.

The distance between two instances x_i and x_j is defined as $d(x_i, x_j)$, as shown in formula (2.4).

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} \left(a_r(x_i) - a_r(x_j)\right)^2} \tag{2.4}$$

The distance between two samples indicates their similarity, and the K samples that are closest to each other in the space are classified into one category. KNN can be used for both classification and regression, but because it requires a lot of space to save training examples, the classification overhead is relatively large. The KNN algorithm also has some improved algorithms, such as the distance-weighted KNN, which weights the sample contributions of K neighbors, and assigns larger weights to the nearest neighbors.

2.3.3 Naive Bayes

Bayesian algorithm is a collective term for a series of classification algorithms based on Bayesian probability law [7]. According to the different prior probability in Bayesian formula, it can be divided into Maximum Likelihood Model, Polynomial model, Poisson model, etc. The characteristic of Bayesian algorithm is that each observed training sample can incrementally reduce or increase the estimated probability of a certain hypothesis. Prior knowledge can be combined with the observed sample data to determine the final estimated probability of the hypothesis. Bayesian Algorithms allow assumptions to make uncertain predictions.

Bayesian formula is the basis of Bayesian algorithms, because it provides a method to calculate posterior probability $P(h|D)$ using prior probabilities $P(h)$ and $P(D)$, as shown in formula (2.5).

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \tag{2.5}$$

Among the Bayesian learning methods, the most practical and widely used is the Naive Bayes Classifier, whose fitting effect is comparable to some neural networks. The word naive in Naive Bayes means that the premise of the probabilistic classifier is relatively simple. The simple point is that it assumes that each sample is conditionally independent of each other. Giving the naive Bayes classifier a series of training examples about the objective function $f(x)$ and new examples $\langle a_1, a_2, \dots, a_n \rangle$, and then predict the target value or category of these new examples.

The goal of the new example classification of Bayesian method is to obtain the maximum target value v_{MAP}, as shown in formula (2.6).

$$v_{MAP} = \arg\max_{v_j \in V} P\left(v_j | a_1, a_2, \dots, a_n\right) \tag{2.6}$$

This expression is transformed by Bayesian formula, as shown in formula (2.7).

$$v_{MAP} = \operatorname{argmax}_{v_j \in V} \frac{P\left(a_1, a_2, \dots, a_n | v_j\right) P\left(v_j\right)}{P\left(a_1, a_2, \dots, a_n\right)} \tag{2.7}$$

What the classifier needs to complete is based on the values of the two data items in the numerator of formula (2.7). It is estimated that each sample probability $P\left(v_j\right)$ only needs to calculate the probability of the target value v_j in the training sample. Then we assume that all attribute values are conditionally independent of each other, that means,

the observed probability of $a_1, a_2, \ldots, a_n$ is equal to the product of the probabilities of each individual attribute. So, the Naive Bayesian method can be obtained from formula (2.7), as shown in formula (2.8).

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} \prod_j P(a_i|v_i) \tag{2.8}$$

In summary, the Naive Bayesian learning method needs to calculate the statistical probabilities $P(a_i|v_i)$ and $P(v_j)$ of different examples on the training data. And finally use formula (2.8) to do the classification.

2.4 Deep Learning Classification Method

Deep learning methods generally refer to deep neural networks based learning algorithms. Deep neural networks can be mainly divided into CNN, RNN, etc. [29]. In recent years, deep learning algorithms have been widely used in text classification tasks. These methods first map the words into vectors, and then extract the features in the word vectors and classify them. When the neural network has much more layers, the extracted features can be more typical. The model is usually trained on large-scale data sets.

The neural network algorithm actually belongs to a kind of machine learning algorithm, which refers to a model that uses multiple artificial neural nodes to fit a complex function [30]. Neural network algorithms are inspired by the human brain nervous system and can model the correspondence between data. Early neural networks were not used in the field of machine learning, but were used to fit complex functions.

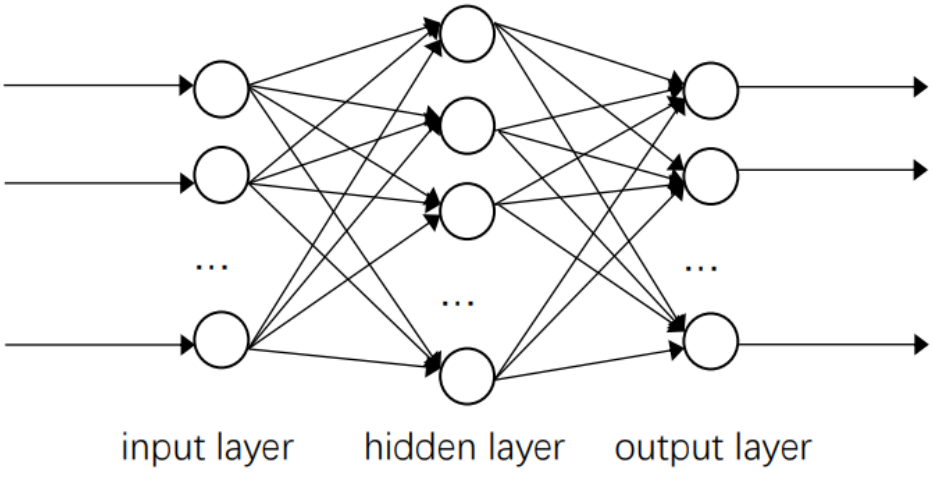

Figure 2.4 Basic structure of neural network

The structure of the artificial neural network is shown in Figure 2.4. It mainly includes an input layer, some hidden layers and an output layer. Each layer has multiple artificial neural network nodes, and each node has a weight representing the degree of influence to the next node. These nodes also have a non-linear function as the activation function, which is used to recalculate the weight. The final output is the extracted feature or the compressed information. If it is a classification or prediction model, use a probability distribution function (such as Softmax) in the output layer to normalize the calculated probability vector to obtain the distribution probability on each label, and take the one dimension with the largest value as the final prediction result.

The deep neural network, as its name implies, is a neural network with a large number of layers, but when the number of layers increases, the problem of gradient explosion will occur [31]. Around 1986, after the proposal of the layer-by-layer differential back-propagation algorithm, which could alleviate the gradient explosion problem, the deep neural networks started to become popular. However, the problem of the disappearance of the gradient still exists. The usual solution is to assign values to the nodes of each layer of neural network through unsupervised learning, so that we can accurately predict the output of the next layer, and fine-tune through the backpropagation algorithm. The method of pre-training and fine-tuning effectively reduces the difficulty of training procedure of deep neural network, and has achieved great success in speech recognition and image recognition.

2.4.1 Convolutional Neural Network

CNN is a Feedforward Neural Network based on convolutional calculations with a deep network structure. CNN was proposed by LcCun et al. [32] in the 1980s and used in the image field in the early days, its basic structure is convolutional layer plus pooling layer. In NLP field, CNN was firstly applied in relation classification and sentence classification tasks. Because CNN can directly deal with images and avoid various complex pre-processing steps, it has been widely used [33]. From the 1980s to the 1990s, CNN developed rapidly, such as the proposal of LeNet and Time Delay Networks. On the basis of CNN, He et al. [34] proposed a residual structure, which transfers the gradient of the previous layer to the next layer through additional mapping, which greatly alleviates the problem of gradient attenuation and greatly enhances the

performance of CNN. By applying this method, we can increase the number of convolutional layers and extract deeper features.

The common CNN structure generally consists of five types of layers, the details are as follows:

(1) Input layer

The role of the input layer is to reduce data dimensionality. Generally, the two-dimensional CNN is used to process image data, and the one-dimensional CNN is used to process text sequence data. Since the back propagation in the neural network may cause the gradient explosion problem, the CNN model needs to use the gradient descent method to update the parameters to alleviate the problem. The data received by the input layer generally needs to be normalized to improve the computational performance of the model.

(2) Convolutional layer

The role of the convolutional layer is to extract features from the input data. Each convolutional layer of CNN is composed of several convolution kernels. Each kernel is responsible for extracting different features. Through the extraction of sliding window, local features of text data can be learned. The model parameters are generally updated iteratively using backpropagation algorithms.

(3) Activation layer

The role of the activation layer is to implement the nonlinearity of the model. The activation layer is generally behind the convolution layer, using the activation function to calculate the convolution neuron, such as sigmoid function, ReLU function, and tanh function. At present, the commonly used activation function in CNN is ReLU function, because it can converge fast and easily calculate the gradient.

(4) Pooling layer

The role of the pooling layer is to compress the feature dimension. Generally, the feature vectors obtained by the convolutional layer have a large dimension. Therefore, we set the pooling layer in the middle of convolutional layers, divide the features into several regions and obtain the lower-dimensional features. Pooling operations include max pooling, average pooling, etc. Pooling operations can effectively prevent overfitting.

(5) Fully-connected layer

The fully connected layer is generally set at the end of the CNN, which is equivalent to the hidden layer in the feedforward neural network. Through the fully connected layer, CNN aggregates all the local features extracted by the convolutional layer and the

pooling layer into global features, and calculates the weight score of each category.

The main features of CNN are: 1) Local area perception. The neurons between each layer in CNN are not fully connected, but partially connected. Each neuron only feels some local features, and then in the highest convolutional layer, the features sensed in these local areas can be aggregated to global features; 2) Weight sharing. The weight parameters of each neuron are the same as the local area, sharing the same convolution kernel. But one convolution kernel can only learn one type of features, generally it is necessary to design multiple convolution kernels to learn more features; 3) Downsampling of data in time or space. Because of the local area perception feature, CNN needs to reduce the amount of data processing to retain useful feature information.

2.4.2 Recurrent Neural Network

RNN is improved from the simple recurrent network proposed by Elman et al. [35], and has wide applications in NLP. RNN is a network with short-term memory capability. It is usually used to process data with timing characteristics. It is mainly composed of cyclic coding units for processing sequences. The structure of cyclic coding units mainly includes input layer, hidden layer, and output layer. In RNN, each neural node may not accept the information of other ganglia, or it may only accept the information of the previous neural node, forming an effective closed loop. The internal structure of RNN can be seen as multiple copies or cycles of the same neuron. Layers are connected by weights, and the output is controlled by activation functions. Each neuron module C transmits information to the next neuron. The network structure of RNN is shown in Figure 2.5 [49]

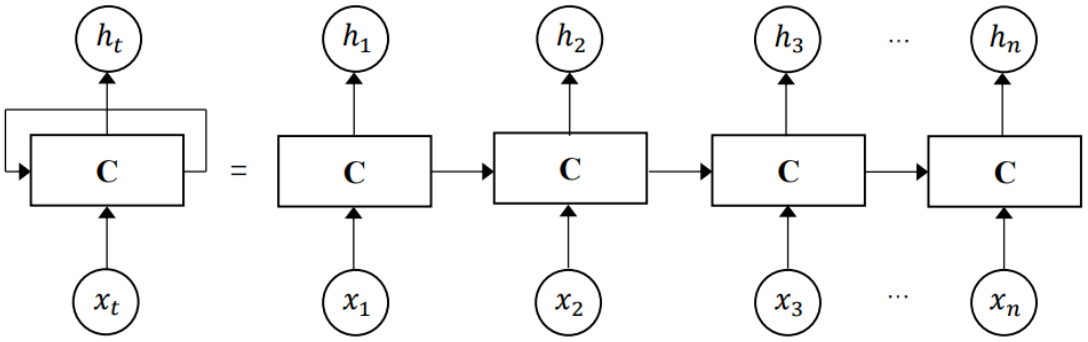

Figure 2.5 The structure of RNN

RNN uses the back-propagation algorithm over time for training and learning, but

when the data sequence is long, the returned residual index decreases, and the neuron weight update is slow, which will cause long-term dependence problems, resulting in poor model performance. In order to preserve the long-term memory of the sequence, it is necessary to introduce a storage unit, so many improved models of RNN are proposed, such as LSTM [36], Gate Recurrent Unit (GRU) ". LSTM network belongs to a special RNN network, it can learn long-term dependent features, which is more suitable for learning and predicting important information with relatively long delays or intervals in time series. Gers et al. "for the first-time applied LSTM to text classification, which is a significant improvement over machine learning algorithms. Furthermore, Bi-LSTM has also been introduced into the field of text processing. Its characteristic is that it cannot only extract the features of the previous text, but also obtain the features of the subsequent sequence.

In the traditional RNN neural network, the repeated neuron module only contains a simple structure, such as a single activated tanh layer. LSTM also has repeated neuron modules, but the modules have a different structure from RNN. The LSTM neural network unit module contains three gates, namely input gate, forget gate and output gate, which are used for the retention and control of characteristic information, so as to learn and remember long-distance text information. The structure of LSTM is shown in Figure 2.6 [49].

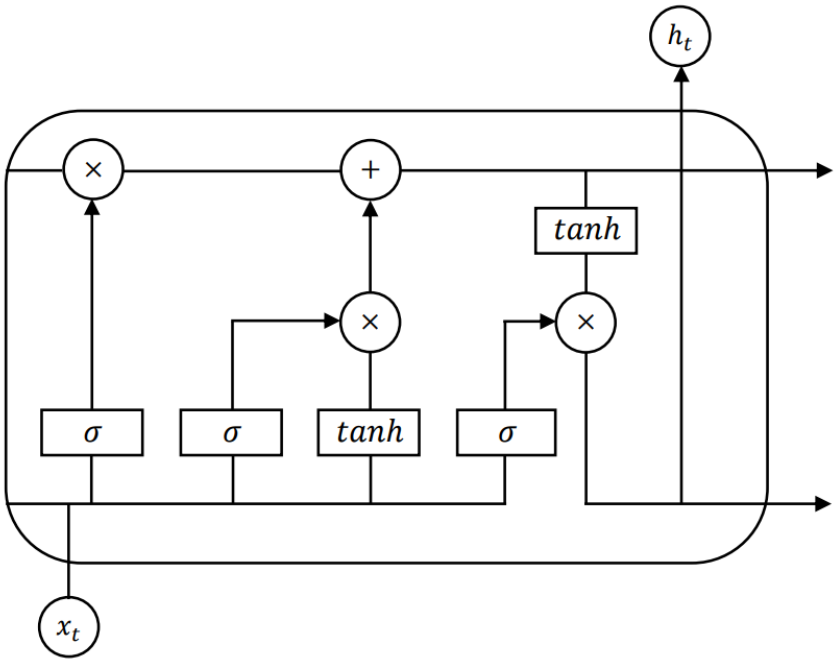

Figure 2.6 The structure of LSTM

According to the structure of LSTM network, the calculation of each LSTM is shown

in formulas (2.9) ~ (2.14).

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{2.9}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2.10}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{2.11}$$

$$\check{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{2.12}$$

$$C_t = f_t * C_{t-1} + i_t * \check{C}_t \tag{2.13}$$

$$h_t = o_t * tanh(C_t) \tag{2.14}$$

In the formula, f_t, i_t, o_t represent the forget gate, input gate, and output gate, $\check{C}_t$ represents the state of the LSTM unit at the previous moment, C_t represents the state of the current LSTM unit, h_{t-1} represents the output of the previous unit, h_t represents the output of the current unit.

By adding three control units, LSTM makes the weights in the loop process iterative, which can effectively avoid gradient disappearance or gradient explosion. The training algorithm of LSTM is still the back propagation algorithm.

2.5 Other Classification Methods

In addition to the common models like CNN and RNN, deep learning classification algorithms have many other forms, such as Attention model, Pre-training Language model, etc. These methods are flexible in form and variable in structure. They also applied into a lot of tasks in NLP field.

2.5.1 Attention Model

In recent years, attention-based models have been widely used in NLP. LSTM+Att Translation model [39], using LSTM as the encoder, the attention mechanism saves the feature vector, effectively avoiding the loss of features from the encoder to the decoder. Att+Tree model [40] uses attention to extract features such as the semantic information of the syntax tree in the text. On the basis of Bi-LSTM, Sutskever et al. [41] proposed a sequence-to-sequence model that combines the attention mechanism, which has been successfully applied to multiple tasks including text classification. Zhang et al. [42]

proposed Attention Pooling-Based Convolution Neural Network (APCNN) that combines the attention mechanism with CNN, which applies the attention mechanism to the pooling layer and effectively solves the problem of information loss in the pooling layer.

2.5.2 Pre-training Language Model

Following with the various attention models, the pre-training language model was proposed around 2017. It is a new type of deep learning model. These pre-training models are based on deep neural networks, attention mechanisms, language models, and get very good results in many NLP tasks, typical representatives include ELMo, BERT, etc. This type of model uses more training data than previous deep learning methods, and uses language models to dynamically train word vectors, which solves the ambiguity of polysemous words in different contexts. Among them, the GPT proposed by Radford et al. [11] uses LSTM to build language model for predicting word vectors based on dynamic context. The ELMo model proposed by Peters et al. [12] is based on GPT but uses bidirectional LSTM coding to construct language model, which can generate deep vector representations. After that, BERT [13] is characterized by only using Transformer as the node of the language model, it performs better than GPT and ELMo, but also cost much more amount of data and training resources. Since 2019, XLNet [14] takes the first place in many public classification tasks. Pre-training language models developed rapidly and can be widely used in all NLP field. In addition to text classification tasks, they can also be applied to tasks such as text vector representation, text generation, and translation.

2.5.3 Other Models

Deep learning methods, including neural network models combined with attention, can automatically learn features from large-scale data sets, but have some common problems in interpretability and convergence time. [43]. Considering the complexity of deep learning models, some more simplified models has been proposed. They can also get good results on large-scale data sets. The FastText model proposed by Joulin et al.

[44] is based on the CBOW language model, which constructs a non-linear classifier based on word vector average. Shen et al. [45] proposed the Simple Word Embedding Model (SWEM). This method is based on the maximum pooling and average pooling of word vectors without using deep neural networks. Vaswani et al. [10] proposed a generative model named Transformer, which uses multiple parallel attention layers to do the encoding, and has achieved a breakthrough in machine translation, verifying that the coding effect of the attention mechanism is better than LSTM. Shen et al. proposed a classification model that using two-way self-attention mechanism named Directional Self-attention Network (DiSAN), achieved good results on multiple data sets such as text classification and text inference. The model did not use CNN and RNN structures, it only included the feedforward neural network.

Chapter 3 Proposed Approach

Aiming at the shortcomings of the current standard deep learning algorithm training rate and the shortcomings of the accuracy of Chinese multi-label text classification, we propose a Chinese multi-label text classification method based on pre-training language model, attention mechanism and pooling mechanism. The attention part of the model can focus on the more important local features of the text data itself, and the pooling part can obtain the overall features by directly operating the word vector.

3.1 Attention Mechanism

The attention mechanism is derived from the process of human's attention to certain specific things. It was first applied in the image field to focus on the more important parts of an image, and later applied to the NLP field to extract more noteworthy parts from a text, and get deeper semantic information.

The mathematical representation of the attention mechanism can be described as the process of matching a query Q with a series of key-value pairs (K, V) and outputting. As shown in formula (3.1).

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.1}$$

In the formula, Softmax is the normalization function, Q is the query vector, K is the key vector, V is the value vector, and d_k represents the vector distance between Q and K.

The general structure of the attention mechanism is shown in Figure 3.1. In the figure, the query Q and the key vector K calculate the similarity between the two through the scoring function s. The scoring function s can use cosine similarity, matrix multiplication, and so on. Then input the result of the similarity into the normalization function Softmax, and then output it after normalization. Then sum the value V for vector multiplication. Finally, concatenate the results of each operation to get the attention weight a, which is the attention a of Q to K. This structure can also be called the key-value pair mode of the attention mechanism, that is, the attention is calculated by

querying the operation of Q on the key-value pair (K, V), which is usually applied to end-to-end tasks such as machine translation and reading comprehension.

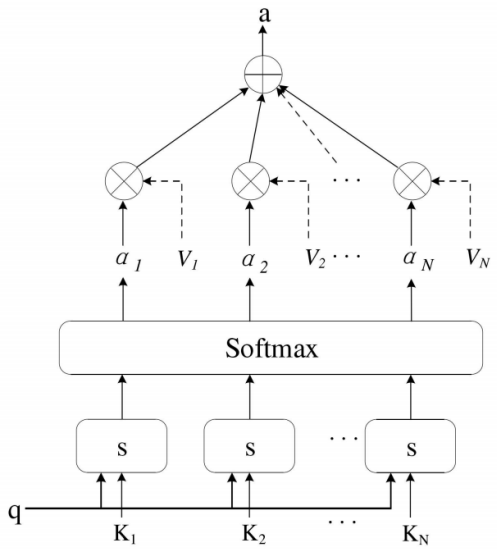

Figure 3.1 Attention mechanism

When Q=K=V, the key, value, and query are the same vector, it is a self-attention mechanism, as shown in Figure 3.2. In the figure, the operation of querying Q and key vector K is consistent with the normal mode. In the multiplication calculation after Softmax, the same vector AND as K is used. Calculation, the calculation process before and after has not changed. The self-attention mechanism is equivalent to performing attention operations on the feature dimensions inside the text vector. In natural language processing, this structure is usually applied to tasks such as text classification, text labeling, and model pre-training.

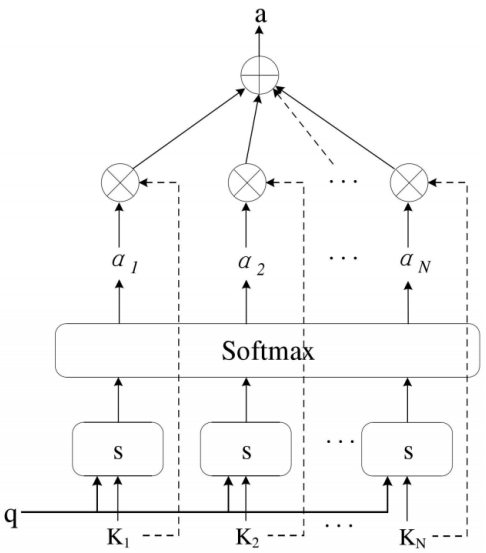

Figure 3.2 Self-attention mechanism

There are also variants of the attention mechanism such as multi-head attention and two-way attention. Multi-head attention is represented by the Transformer model proposed by Vaswani et al. in 2017 [10]. This model uses a combination of a five-layer multi-head attention mechanism and a self-attention mechanism to significantly improve the accuracy of neural machine translation. In 2018, DiSAN proposed by Shen et al. [46] uses two-way attention and self-attention, and achieves great results in tasks such as text classification.

3.2 Attention-Pooling Text Classification Model Based On BERT

In view of the good performance of the attention model in text classification, we propose Attention-Pooling Text Classification Model Based On BERT. The model mainly consists of three parts: word embedding layer, pooling layer and attention layer. Inspired by the pooling layer in the CNN, we mainly use maximum pooling for feature extraction.

The model structure is shown in Figure 3.3, where dense represents the fully-connected layer. w represents the word in the text, $(\omega_1, \omega_2, \cdots, \omega_n)$ represents word vectors, and the softmax in the last represents the classification function. The detail of each part will be described below.

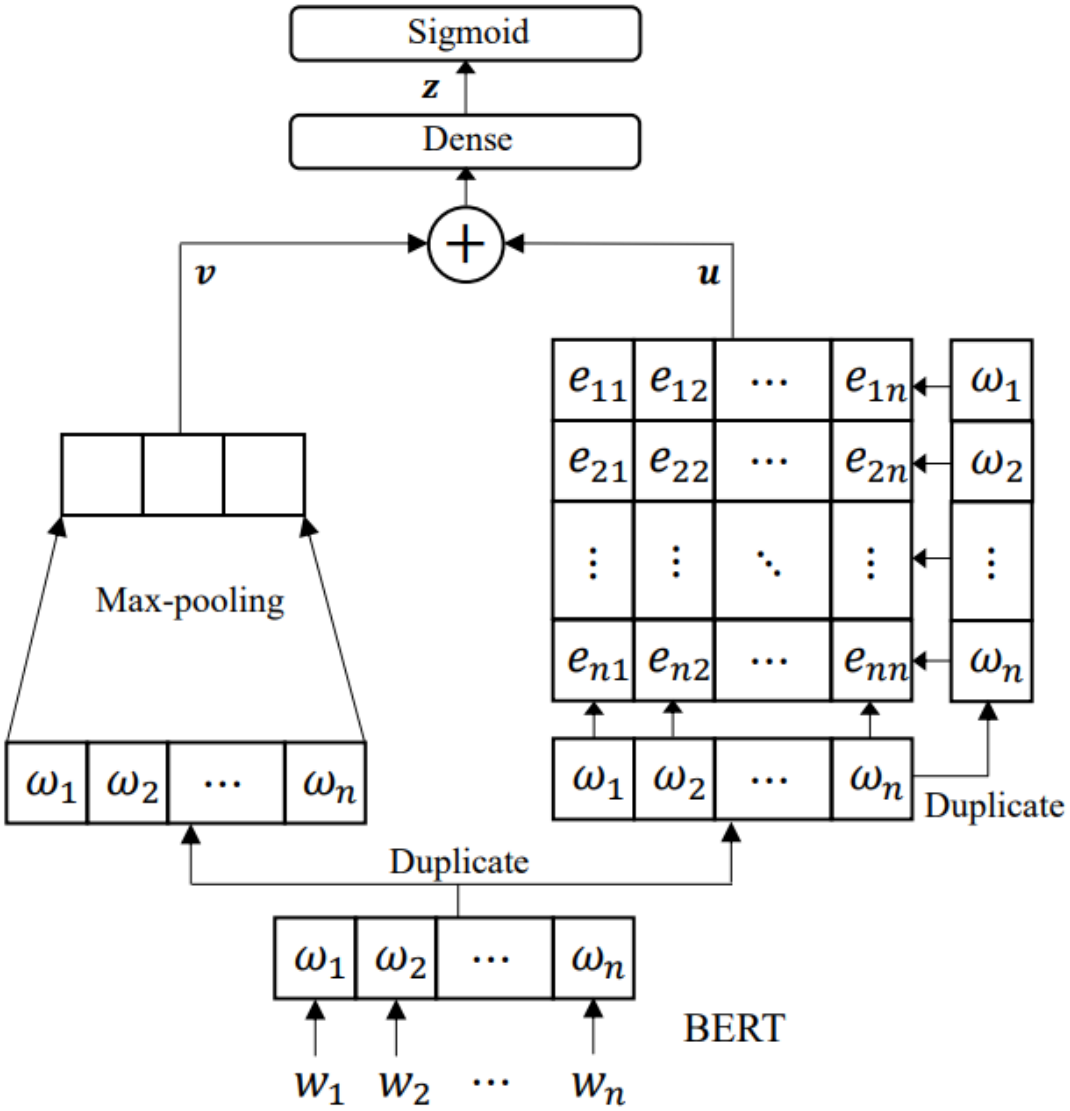

Figure 3.3 Structure of the model

3.2.1 Word Embedding Layer

In the process of text classification, the level of data quality directly affects the classification effect. As the basic unit of text data, words store the basic characteristics of the text. In addition, part-of-speech information and word position information also have a certain guiding significance for judging the text category.

Word embedding is a distributed vector representation of vocabulary, and it is also a form of encoding. It can effectively encode the meaning or semantic information network of a vocabulary in a high-dimensional space. This paper uses the pre-trained language model BERT to convert words into word vectors.

3.2.2 Attention Layer

The function of the attention layer is to extract important local features and to focus on feature vectors that are more representative of the overall text sequence. In the attention layer, firstly, given a query text vector Q, find out the feature vectors in the vector matrix that are more related to the query vector Q, and filter out irrelevant features, so as to greatly shorten the time of feature extraction and speed up the training of the model. Self-attention can focus on the word or vector with the highest weight within the text, and get the weight distribution of a series of words, and finally merge into one attention weight.

3.2.3 Pooling Layer

The role of the pooling layer is to extract global features, which can directly extract and compress the weights of each dimension in the word vector feature sequence. This layer mainly performs Max-Pooling operations. The key to the pooling operation is to operate on the same dimension of each feature vector in the entire text sequence to achieve feature dimensionality reduction. Since each dimension in the vector feature of a word contains different semantic information, the pooling operation integrates the information on each dimension to obtain the global information on the entire text.

3.2.4 Prediction layer

The prediction layer is the layer that can output the final classification result. This layer mainly uses the Softmax function to obtain the final probability distribution result. Softmax function is also called the normalized exponential function, which is to divide all the dimensions of a vector according to the value and expressed it as a probability distribution, and its sum is 1, as shown in formula (3.2).

$$Softmax(x_j) = \frac{exp(x_j)}{\sum_{k=1}^{K} exp(x_k)} \tag{3.2}$$

But in our case, in order to feed the requirement of multi-label classification, we choose the Sigmoid function, as shown in formula (3.3).

$$Sigmoid(x) = \frac{1}{1+exp(-x)} \tag{3.3}$$

In the case of multi- classification, sigmoid allows processing non-exclusive labels (also called multi-labels), while softmax deals with exclusive classes.

Chapter 4 Experiments and results

In order to train the model, we established a small-scale Chinese multi-label text data set. For verifying the performance of the model, we also used the data set to train other networks, and compared the output of each network.

4.1 Data Collection

We have established a small-scale Chinese multi-label text data set, and the source of the corpus is the Chinese Internet forum. The reason for choosing this type of text is that there are many and complex speeches in the forum, and valuable information is often mixed with a lot of meaningless information. Even with the current search engine technology, it is difficult to filter out high-value information. This characteristic is very suitable for verifying model performance.

This data set contains 5000 texts of various lengths and 10 labels. We use manual labeling to give relevance to text and labels. At the same time, we have also performed batch preprocessing on all texts, such as replacing hyperlinks in the text with ordinary text, deleting emoji characters in the text, etc.

4.2 Training Detail

The hyperparameters of this model are set as follows: the dropout rate is 0.35, the learning rate of the fully connected layer is 0.003, and the batch size is 64. The Adam optimizer optimization algorithm is used in the training process, the loss function uses the cross-function function, and the early stopping strategy is used to stop training after the accuracy rate has stabilized after a certain cycle batch to prevent overfitting. The model is trained on the training set and validation set to achieve the best accuracy and then tested on the test set. The training environment is Python 3.7 and the OS is Windows 10. The model is implemented in Tensorflow-GPU-1.14.0 with 8G GPU memory and 16G memory. The main hyperparameters are shown in Table 4.1.

Table 4.1 Main hyperparameters

Hyperparameter	Setting value
Learning rate	0.003
Dropout	0.3
Batch size	64

The dropout rate is the proportion of the number of randomly discarded nodes in the neural network. Using the dropout rate can prevent the model from overfitting to the training data.

In order to find out the effect of dropout rate on accuracy, we set different dropout rates on the train set and test set. The experiment results are shown in Figure 4.1. Judging from the accuracy changes on the train set and the test set, when the dropout rate is less than 0.25, the accuracy on the train set is low, indicating that the training is insufficient and the model is under-fitting. When the dropout rate is greater than 0.25, the accuracy on the train set steadily increases, but the accuracy on the test set has reached an inflection point, indicating that the overfitting occurs. Therefore, the final dropout rate is set to 0.25 to make the model fit the training data appropriately. The debugging methods of other hyperparameters are the same.

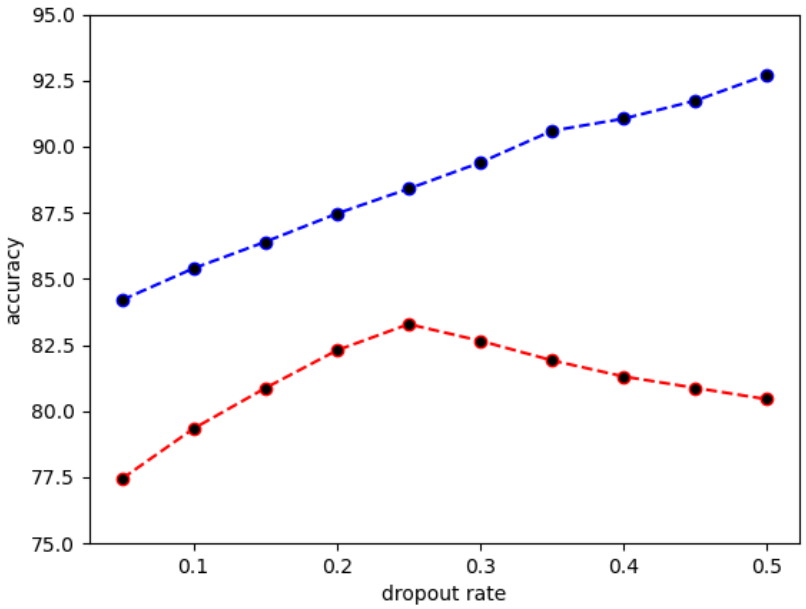

Figure 4.1 The influence of dropout rate on accuracy

4.3 Evaluation Method

The evaluation methods of this paper are Macro Precision Rate (MacroP), Macro Recall Rate (MacroR), and Macro F1 value (MacroF1). MacroP is used to evaluate the classification accuracy of each category of the model. MacroR is used to evaluate the coverage of correct answers during the classification process. For classification models, it is always difficult to increase precision and recall at the same time. So, we can use MacroF1, which is the adjustment and average value of MacroP and MacroR, to evaluate the comprehensive performance of the classification model. The detail is shown in formula (4.1) ~ (4.3).

$$MacroP = \frac{1}{n}\sum_{k=1}^{n}\left(\frac{correct_k}{predict_k} \times 100\%\right) \tag{4.1}$$

$$MacroR = \frac{1}{n}\sum_{k=1}^{n}\left(\frac{correct_k}{true_k} \times 100\%\right) \tag{4.2}$$

$$MacroF1 = \frac{MacroP \times MacroR \times 2}{MacroP + MacroR} \tag{4.3}$$

4.4 Result and Analysis

In order to verify the effectiveness of the proposed model, this paper compares it with CNN, LSTM, and APCNN. Among them, TextCNN and LSTM are used as a comparison of standard deep learning models. APCNN [42] is a model that combines CNN, Bi-LSTM and attention mechanism, as a comparison of deep learning models combined with attention. Different from the proposed model, after extracting text features through CNN and Bi-LSTM, APCNN directly inputs the results of attention calculation to the prediction layer. The comparison results of the accuracy of each model on the test set are shown in Table 4.2.

Table 4.2 Model comparison results

	Accuracy/%	MacroP/%	MacroR/%	MacroF1/%
TextCNN	73.74	74.09	78.03	76.01
LSTM	71.30	70.33	71.98	71.15
APCNN	73.78	73.00	71.12	72.05
Ours	83.29	77.51	75.28	76.38

It can be seen from the table that the model proposed in this paper has an accuracy rate of 83.29% on the test set, which is higher than the neural network model TextCNN,

LSTM and the attention model APCNN. In terms of macro precision, the proposed model is also superior to all other models in the experiment. But in terms of recall rate, it is slightly lower than CNN, because the CNN model also has a pooling layer mechanism, which is suitable for classification tasks. It shows that this model is not particularly prominent in coverage. Also see that the accuracy of CNN is 73.74%, which is lower than the two attention models. This may be because although CNN can extract local features and then synthesize them into global features, it is difficult to pay attention to the important features of the overall text. The higher MacroFl value of the proposed model indicates that it performs well overall. The accuracy of the LSTM model is 71.30%, indicating that in the case of long text data, although the LSTM can memorize farther information through the control unit, the forget gate in the control unit will also discard the longer time series. Therefore, the memory gate can only learn the characteristic information in a certain time or space sequence. Therefore, when LSTM processes long texts, the accuracy rate will decrease. The accuracy rate of APCNN is 73.78%. The possible reason is that the category distribution on the data set is quite different, which leads to the unsatisfactory classification effect.

Chapter 5 Conclusion

This paper proposes a model based on attention and pooling mechanism. In order to shorten the convergence time, this model does not use a complex deep neural network structure in feature extraction step. Instead, it uses attention to extract relatively important local features of the text, uses pooling to extract the global features of the text, and combines those advantages to apply into text classification tasks.

In order to evaluate and test the performance of the proposed model, a number of experiments have been carried out in this paper. The results of comparative experiments with related models show that the proposed model can not only achieve higher accuracy, but also has certain speed advantages.

Bibliography

[1] Rumellhart D E. Learning internal representations by error propagation [J]. Parallel distributed processing, 1986, 1: 318-362.

[2] Zhang X, Zhao J, LeCun Y. Character-level convolutional networks for text classification[C]//Advances in neural information processing systems. 2015: 649-657.

[3] Bowman S R, Angeli G, Potts C, et al. A large annotated corpus for learning natural language inference[C]//Proceedings of the 2015 conference on empirical methods in natural language processing (EMNLP). 2015: 632-642.

[4] Wang C, Zhang M, Ma S, et al. Automatic online news issue construction in web environment[C]//Proceedings of the 17th international conference on World Wide Web. 2008: 457-466.

[5] Hinton G E, Osindero S, Teh Y W. A Fast Learning Algorithm for Deep Belief Nets [J]. Neural Computation, 2006, 18(7):1527-1554.

[6] Tong S, Koller D. Support vector machine active learning with applications to text classification[J]. The Journal of Machine Learning Research, 2002, 2(1):999-1006.

[7] Mladenic D, Grobelnik M. Feature selection for unbalanced class distribution and naive bayes[C]//Proceedings of the Sixteenth International Conference on Machine Learning. 1999, 99: 258-267.

[8] LeCun Y, Bottou L, Bengio *Y,* et al. Gradient-based learning applied to document recognition [J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[9] Jordan M I, Ghahramani Z, Jaakkola T S, et al. An introduction to variational methods for graphical models [J]. Machine learning, 1999, 37(2): 183-233.

[10] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Proceedings of the 31st Annual Conference on Neural Information Processing Systems. California: NIPS,2017:5998-6008.

[11] Peters M E, Neumann M, Iyyer M, et al. Deep contextualized word representations [C]//Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. New Orleans:ACL,2018:2227-2237.

[12] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding [C]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2019:4171-4186.

[13] Zou X, Sun N, Zhang H, et al. Diachronic Corpus Based Word Semantic Variation and Change Mining[C]//Language Processing and Intelligent Information Systems. Springer, Berlin, Heidelberg, 2013: 145-150.

[14] Socher R, Perelygin A, Wu J, et al. Recursive deep models for semantic compositionality over a sentiment tree-bank[C]//Proceedings of the 2013 conference on empirical methods in natural language processing. 2013: 1631-1642.

[15] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[EB/OL].(2013-l-16)[2020-4-16]. http://arxiv.org/pdf/1301.3781.pdf.

[16] Pennington J, Socher R, Manning C D. Glove: Global vectors for word representation[C]//Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014: 1532-1543.

[17] Sebastiani E Machine learning in automated text categorization[J]. ACM computing surveys (CSUR), 2002, 34(1): 1-47.

[18] Cover T M, Hart P E. Nearest Neighbor Pattern Classification [J]. IEEE Transactions on Information Theory, 1967, 13(1):21-27.

[19] Zhou P, Qi Z, Zheng S, et al. Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling[C]//Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers.2016:3485-3495.

[20] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting[J]. Journal of Machine Learning Research, 2014, 15(1):1929-1958.

[21] Liang B, Li H, Su M, et al. Deep text classification can be fooled[C]//Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence Main track. 2018:4208-4215.

[22] Lecun Y, Boser B, Denker J, et al. Backpropagation Applied to Handwritten Zip Code Recognition [J]. Neural Computation, 1989, 1(4):541-551.

[23] Liu K, Zhang M, Pan Z. Facial expression recognition with CNN ensemble[C]//2016 international conference on cyberworlds (CW). IEEE, 2016: 163-166.

[24] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition.2016: 770-778.

[25] Cleeremans A, Servan-Schreiber D, McClelland J. Finite State Automata and Simple Recurrent Networks [J]. Neural Computation, 1989, 1(3):372-381.

[26] Hochreiter S, Schmidhuber J. Long short-term memory [J]. Neural computation, 1997, 9(8): 1735-1780.

[27] Gers F A, Schmidhuber J, Cummins E Learning to forget: Continual prediction with

LSTM[J]. The Institution of Engineering and Technology, 1999: 850-855.

[28] Yokobori T, linuma H, Shimamura T, et al. Plastin3 Is a Novel Marker for Circulating Tumor Cells Undergoing the Epithelial-Mesenchymal Transition and Is Associated with Colorectal Cancer Prognosis[J]. 2013, 73(7):2059-2069.

[29] Wenfeng Liu, P. Liu, Y. Yang, et al. An Attention-Based Syntax-Tree and Tree-LSTM Model for Sentence Summarization [J]. International Journal of Performability Engineering, 2017, 13(5):775-782.

[30] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[C]//Advances in neural information processing systems. 2014: 3104-3112.

[31] Er M J, Zhang Y, Wang N, et al. Attention pooling-based convolutional neural network for sentence modelling [J]. Information Sciences, 2016, 373: 388-403.

[32] Liu T, Zhu W, Liu G. Research Progress of Text Classification Based on Deep Learning [J]. Electric Power Information and Communication Technology, 2018(3): 1-7.

[33] Joulin A, Grave E, Bojanowski P, et al. Bag of tricks for efficient text classification[C]// Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics. 2017:427-431.

[34] Shen D, Wang G, Wang W, et al. Baseline needs more love: On simple word-embedding-based models and associated pool-ing mechanisms [C]//Proc of the 56th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA:ACL, 2018: 440-450.

[35] Shen T, Zhou T, Long G, et al. Disan: Directional self-attention network for rnn/cnn-free language understanding[C]//Thirty-Second AAAI Conference on Artificial Intelligence. 2018:5446-5455.

[36] Li S, Zhao Z, Hu R, et al. Analogical reasoning on Chinese morphological and semantic relations [C]//Proc of the 56th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA:ACL, 2018: 138-143.