Affective Analysis of Chinese Sentences Based on Word2vec and SVC

Fu-yong WAN$^1$* and Shi-qiang LI$^2$

$^1$School of Mathematical Sciences, East China Normal University, Shanghai, China
$^2$Shanghai Dataseed Information Technology Co., Ltd., Shanghai, China

*Corresponding author

Keywords: Chinese text classification, Word2vec, SVC, Affective analysis.

Abstract. This Paper was based on word2vec, Using Chinese Encyclopedia Corpus of Wikipedia as the training set to generate Chinese word vector and Chinese sentence vector, and using SVC (support vector classification) to classify the text of 16542 comments in a hotel industry, that is, to realize the affective analysis of Chinese sentences. The results show that the sentence vectors generated by the voting SVC model are better than those generated by the mean SVC model.

Introduction

The Internet of big data era contains a great deal of information and data, mainly text, sound, image and video. The artificial intelligence technology of the new era coming along with big data, including two main directions: computer vision and natural language processing. Text classification is the basic task of natural language processing science. The text here can be news, mail, web page, book, voice text or part of it.

Classifying text information automatically according to the content of text can help people to organize text, mine information, locate information accurately and distribute information. Text categorization is ubiquitous in the Internet era, advertising short message shielding, web search and positioning can not be separated from text classification [1]. Studying text classification algorithm and improving text classification accuracy can greatly reduce labor cost in actual production and make artificial intelligence better serve people's lives [2].

Foreign research on automatic text classification was carried out earlier, in the late 1950s, The method of automatic text classification based on word frequency statistics has made a groundbreaking breakthrough in this field. In 1960, Maron published the first paper on automatic classification algorithm. Subsequently, many scholars, such as K.Spark and K.S.Jones, have also carried out fruitful research work in this field. At present, the research on text classification in foreign countries has entered from the experimental stage to the practical stage. And it has been widely used in mail classification, electronic meeting and so on. Among them, the email classification system developed by MIT for the White House is more successful.

Compared with English text classification, one of the important differences in Chinese text classification is the preprocessing stage: Chinese text reading needs to cut words, unlike English words, there are spaces to distinguish them. For a long time, there are no published data sets in the research of Chinese text classification, which makes it difficult to compare the classification algorithms. However, the current Chinese word cutting technology has matured.

In recent years, with Google opening up the source code of its translation system, word2vec methods and neural network algorithms have begun to pour into the field of natural language processing. Among them, with RNN as the main representative, major technology companies are scrambling to develop new natural language processing tools. The intelligence and accuracy of text classification are higher and higher.
The Data of Chinese Corpus and Its Preprocessing

The Data of Chinese Corpus

The data of this paper are 16542 comments of a hotel industry. These data are some comments and comments about the hotel, and guest who live in the hotel. In this paper, there are only three types of comment data, namely, "bad comment" (marked 0), "good comment" (marked 1) and "middle comment" (marked 2). Some of the sample data are shown in Figure 1.

![Figure 1. Raw sample data.](image1)

Chinese Word Segmentation, Punctuation and Deactivation

The first task of dealing with Chinese sentences is Chinese word segmentation. Chinese word segmentation is the first task in natural language processing, which is very important but not easy to implement. The Chinese word cutting in China is better done by "jieba" of the University of Science and Technology of China and the system of cutting words by the Harbin Institute of Technology. The second leading task is to remove punctuation marks from natural languages. This is not difficult, after all, there are not many types of punctuation. The third preposition is to remove stop words such as "啊" and "嗯". The set of deactivated words used in this paper is a list of 1281 stop words in the process of natural language processing. After Chinese word segmentation, punctuation and stop word are removed from the sample data, some of the results are shown in Figure 2. In this case, there is no longer a formal difference between Chinese problems and English problems, and word2vec can be directly used to transform words into word vectors.

Generating Word Vector with Word2vec

Word2vec, an open source neural network algorithm used by Google in 2013 to generate word vectors, was first proposed by its author, Tomas Mikolov, in his paper [3]. But there is no specific algorithm details. Word2vec uses two important models-CBOW model and Skip-gram model. The original author gives two sets of frameworks. It is the design based on Hierarchical Softmax framework in paper [3] and the design based on Negative Sampling frame in paper [4]. The input layer of the neural network algorithm is the word word, output layer corresponding to a Huffman tree, thus forming an output vector vector.

Using the word2vec tool in python's gensim third-party software package, the corpus can be quickly trained and the corresponding word vectors can be generated. Using word2vec to train 16542 comment sentences, we can get the word vector space of all the words in these sentences.

A word vector space consisting of 9678 word vectors is obtained, which is distributed in 200 dimensional Euclidean vector spaces. The similarity of the trained word vectors can be investigated by using the angle cosine of the vectors. For example: to see the similarity between "好" and "不好" in this word vector space, and the similarity between "不好" and "糟糕". The similarity between "干净" and "脏". Note that because the training of word2vec stops at the threshold, the absolute results of each training are not exactly the same for the same corpus, but the relative distribution of word vectors will not change. So it's better to train word vectors and save them instead of training them every time you use them. Figure 3 below shows some of the results of a training session.

The similarity between "干净" and "整洁", "不好" and "糟糕" is very high, which is really the result of this article. But note that "好" and "糟糕", "干净" and "脏" are also very similar! The main
The reason for this problem is that the quality of word vector is not high. This is because the training sample of the model is too small, with only 16542 sentences, which is insignificant for the big data model like word2vec. Word2vec initially set up a word vector for each word at random. Every time a word appears in the text, the word is trained once in its context. If the sample is too small and a word appears only once or twice, then the word vector is only trained once or twice, without adequate training, and the quality of the word vector is naturally not high. All word vectors are not far from initialized word vectors. So you can see the result in Figure 3, these words are very close, and there is no distinction. Therefore, in order to improve the quality of word vectors, we must increase the sample size, let the words that need appear enough times in different sentences in the corpus, let these words be trained repeatedly to obtain high-quality word vector space.

The Combination of Common Sense Word Vector and Context Word Vector

In view of the above question, now continue to increase the sample size of the corpus. Note that training corpus samples are needed to generate word vectors instead of tagged comments so there is no need for manual tagging and simply increasing sample sizes is not a very difficult task. For example, a novel can be added to the model as a corpus at will. In this paper, two corpora are used to improve the quality of word vectors.

The first corpus is a large general-purpose corpus, which is called common sense corpus in this paper. The Chinese Wikipedia encyclopedia is used as a corpus, which is more than 1 GB in size and contains a lot of common sense information. After all, Wikipedia itself is an encyclopedic corpus. Another corpus is called contextual corpus, and there are some differences in natural language in different contexts. A word vector trained by a common sense corpus is called a common sense word vector. It is a word vector trained by a encyclopedic corpus and a word vector space composed of almost all common words in peoples daily life. It contains more than 910,000 word vectors; The word vector trained by context corpus is called context word vector. It is a word vector space that contains the terms of practical problems to be studied in this paper. This part of the corpus is all the comment information about a hotel industry. The file size is more than 50 MB, and more than 130,000 word vectors are trained in the word vector space.

Now all you have to do is connect these two word vector spaces together. The method used in this paper is the concatenation of word vectors. First, the common sense corpus is trained, the word vector dimension is chosen 150 dimensions, then the context corpus is trained, and the word vector dimension is 50 dimension. Then all the words contained in the common sense corpus and the context corpus are assembled together to form a 200 dimensional word vector. In this way, the front part of the word vector contains common sense information, the second half contains context information, and the way the similarity is calculated in this paper is the angle cosine of the vector. Therefore, only common sense and context are very similar words will get a high degree of similarity. As for the words in the context corpus which do not appear in the common sense corpus, they may be technical terms or rare words, and this part of the word vector is stitched with the 150 dimensional zero vector. Through the above operation, the final word vector space is obtained, which contains 130,553 word vectors. Now, take a look at the effect of the newly generated word vector, as shown in Figure 4.

Now, the quality of trained word vectors has improved. Although the similarity between "不好" and "糟糕", "干净" and "整洁" is less than 0.999, the similarity between "干净" and "脏", "好" and "糟糕" is now even smaller. It is less than 0.500. The most important thing is that the difference of similarity can be seen from the result. In other words, the word vector space has a certain degree of
discrimination for words with different meanings. As a matter of common sense, "不好" and "糟糕" are not so close to each other. Although they are derogatory, "糟糕" is obviously more serious than "不好". As for words like "好" and "不好" there is no distinction between word2vec itself. Because word2vec focuses more on finding the semantics of a word from the context, that is to say, it pays more attention to the function and meaning of a word in the grammatical structure. However, the words such as "好" and "不好", although one is positive and the other is derogatory, but there is no evaluation of things, simply looking at the two words, the grammatical structure is actually quite close. For example, a good thing, if "好" is replaced directly by "不好", there is no grammatical problem, but the meaning has changed. Problems like this are inevitable in classic word2vec. The trained word vectors can achieve the distinction shown in Figure 4.

This article is more concerned with the words "干净", "整洁" and "脏" because they are not as ubiquitous as "好" and "不好", and they appear more frequently in statements describing facilities such as rooms. This is exactly what this experiment needs. Here, the similarity between "干净" and "整洁" is very high, because in natural language, "干净" and "整洁" always give a feeling complementary. "干净" and "脏" usually describe hygiene, and the antonyms are very well differentiated in the concatenated word vector space.

**Emotion Analysis Based on Support Vector Machine**

The algorithm flow of emotion classification using support vector machine is shown in Figure 5.

![Figure 5. The algorithm flow of emotion classification using SVM.](image)

With proper word vectors, natural language processing problems can be transformed into mathematical classification problems. In this paper, we use support vector machine (SVM) to classify the experimental texts. As a classical machine learning classifier, SVM can separate two different samples at maximum intervals. However, as mentioned above, word vectors now have a certain degree of differentiation and present different distributions in vector space, so naturally SVM can be used for classification training. But now there are just word vectors, and what this paper really wants to distinguish is sentences. For text classification based on SVM, the input samples are sentence vectors, and labels can be labeled directly.

**The Difficulty from Words to Sentences**

As shown in Figure 5, the steps for generating word vectors on the left are explained earlier, and the classic SVM classification is on the right. So the only thing this algorithm doesn't talk about is the middle step: from word vector to sentence vector, and that's the difficulty of the algorithm. Word2vec word vectors are word vectors trained according to context information, after stitching. Word vector contains common sense information and context information. But so far, each word is still a single fight, not a sentence. In this paper, support vector machine is used to classify, the basic unit can only be sentences, so from word to sentence is an inextricable problem.
One of the most natural ideas is that, like natural language, word vectors can be pieced together to form sentence vectors. But there are two problems, one is to do this, each sentence is not the same length, the resulting sentence vector dimension is uneven, can not be compared; Second, sentence vectors can be very long, and because they are spliced according to the grammatical structure of natural languages, the meanings of "我和你" and "你和我" will be much different in vector comparison. Therefore, according to the natural language, the word vector splice into sentence vector, is not a good way. Another natural idea is the way some companies used in the early days of natural language processing: taking the mean value of the word vector contained in a sentence to represent the sentence vector.

It is true that this method is applied in many places because the average sentence vector contains all the word vector information in the sentence and the dimension of each sentence vector is the same. But there are two obvious problems with this method: one is distortion and the other is unrecognizable sequence. Distortion means that the average sentence vector will inevitably lose a lot of information from the sentence, especially when a sentence is very long, because it contains more word vectors, the information will overlap with each other after the average. The sentence vector will be distorted; "unidentifiable order" means: although this method does solve the problem of the order of sentences like "我和你" and "你和我" (note that the sentence vectors of "我和你" and "你和我" in the average method are exactly the same). But the sentence vectors of "我恨你" and "你恨我" are exactly the same! Thus, the sentence vector under this method has little ability to recognize the sentence order, although it can unify the sentences with the same meaning and different expressions. But it also loses the ability to identify sentences that are similar but have a wide range of meanings.

But using the average method is not worthless. One of the advantages is that the dimension of sentence vector is at least the same, and the dimension is the same as the dimension of word vector, so we can realize this method, and we can use the vector angle cosine to calculate the similarity of sentence vector. Using spliced word vectors, 16542 samples are transformed into sentence vectors by averaging. The similarity of some sentence vectors is shown in Figure 6.

Figure 6. The similarity of some sentence vectors by using the average method.

In addition, from the results shown in Figure 6, one of the advantages of the sentence vector expressed by the average is that it still has a certain degree of discrimination to the sentence, but its ability to distinguish the sentence similarity is relatively weak. The example shown in Figure 7 below illustrates this: both sentences say that the room is too dark to read, but the similarity is only 0.722.

Figure 7. The similarity of two long sentence vectors by using the average method.

Word Vector "Vote"

In order to solve the problem of information distortion in long sentences, support vector machine is used to classify emotion. In this paper, the method of "voting" word vectors to extract sentence vectors is used to solve the problem of information distortion in long sentences.

The core idea of word vector "voting" is to select \( M \) words \( (c_1,c_2,\ldots,c_M) \), to make them judges, and then rate sentences one by one. To rate, suppose there is now a sentence \( \{w_1,w_2,\ldots,w_M\} \). For the first judge \( c_1 \), use the word2vec word vector to calculate the similarity between \( c_1 \) and all the words in the sentence \( w_1,w_2,\ldots,w_M \), and then take the maximum value \( c_1 \) to score the sentence as \( s_1 \).
Follow this step and have all the "judges" score the sentence as $s_i$, we get an $M$ vector $[s_1, s_2, \cdots, s_M]$, which can be used as the sentence vector of the sentence.

The advantage is that as long as the word "judge" is representative, long sentences do not cause information loss. There are, of course, two drawbacks to this approach: one is that it still cannot solve the problem of the order in which sentences are used, and the other is that these "judges" are not easy to choose from.

For the sample of 16542 comments in this paper, there are only 9678 words in all, all of which are considered as "judges", and the sentence vector has 9678 dimensions. If the training sample is very large, the dimension of sentence vector will be longer. So how do you choose the word "judge"? There are two ways: one is artificial choice of field keywords; the other is dimension reduction. Manual choice of key words requires a large amount of labor costs and is difficult to keep up with the growing needs of the times. So this paper uses the K-means clustering method without words, that is, selecting a number as the dimension of the last sentence vector in advance, and then according to the generated word vector space, All 9678 words in the training set are selected for clustering. In this way, similar words will be grouped into one category, and any word can be chosen as the "representative" of this cluster, and then the word can be used as the "judge" to vote on the comments in this paper. In this paper, 500 is selected as the dimension of sentence vector, and the result of similarity calculation of the newly generated sentence vector is shown in Figure 8.

From the present results, the similarity between all sentences has been improved, the similarity between "隔音待加强" and "早餐很不好" is 0.785. But don't care about the absolute value of the similarity, but look at the relative value, as long as the sentence that says the same thing is more similar than the sentence that says different things. The point is that the sentence vector does not have any information distortion even when faced with a very long sentence, such as the previous Figure 7, and now the calculation is shown in Figure 9. It can be seen that for sentences with similar meanings and long sentences and a lot of interfering information, the similarity of sentences is raised from about 0.722 to about 0.923, and the similarity is improved a lot.

With the right sentence vectors, the text categorization task is no different from other ordinary vector classification tasks. There are two kinds of vectors in vector space, and support vector machines are used to find the optimal partition. This maximizes the separation of the two types of samples. The remaining task is to select the appropriate kernel function and constantly adjust the parameters so that the classification effect is better. Then the trained classifier can be used as a classifier model in practical application. When a new comment sentence comes in, a 0 / 1-2 classifier is used first. If it is not a neutral evaluation, then a 0-1 classifier is passed to judge whether it is good or bad.

**Affective Analysis Using Support Vector Machines**

Two kinds of sentence vectors are generated, one is the average method, the other is the word vector "vote" method. Now we use these two sentence vectors to carry out classification experiments using SVM, a two-classification model. For a data set with three categories of samples, the method used in this paper is to classify 0 tag (derogatory sentence) and 1 label (commendatory sentence) into one
class. Samples of 2 labels (neutral sentences) are singled out as a class. Because as mentioned earlier, when the sample is labeled manually in this paper, it will only be marked as "0" or "1" when the comment is considered derogatory or positive, but in the actual sample, the comment of its own commendation or derogation is not the majority. So the number of neutral comments is several times the sum of the remaining two types of comments.

Because of this method, 0/1-2 classification is more difficult than 0-1 classification. Limited to space and experimental time, this paper only classifies 0/1-2 classification, and gives the specific experimental results. The sentences with 0/1 tag are classified into the first category, and the sentences with 2 tags are classified into the second category. Then 90% of the sentences are randomly selected from the two types of samples, which are grouped together as training set samples. The rest is used as a test set. This is because the number of samples in the two categories is obviously more, that is, the second type of sample is several times that of the first type, if 90% of all samples are directly selected as training sets, it is possible to draw out the first kind of samples directly, which is disadvantageous to the study of model classification effect on the test set in this paper. So the proportion of two kinds of samples in training set and test set is about the same. By adjusting the parameters of support vector machine, the performance of training set and test set is similar, that is to say, the phenomenon of over-fitting is reduced. Finally, the polynomial kernel function \((x^3 + y + 5)^2\) and the support vector machine with penalty coefficient \(C = 5\) are selected. The results of training the sentences generated by the average method are shown in Figure 10.

Figure 10. The results of the mean SVC model.  
Figure 11. The results of the voting SVC model.

Then the sentence vectors generated by the word vector "vote" method are classified with the same parameters, and the result is shown in Figure 11.

From the result, the classification effect of sentence vector generated by word vector "voting" is better than that of sentence vector generated by word vector "mean". The F1 score increased from 0.90 to 0.93, and the accuracy rate increased from 82% to 87%. Results as shown in Table 1.

| Method            | The accuracy | The recall rate | F1 score |
|-------------------|--------------|-----------------|----------|
| The mean SVC model| 0.82         | 1.00            | 0.90     |
| The voting SVC model | 0.87         | 1.00            | 0.93     |

**Conclusion**

The F1 score shows that the sentence vectors generated by the voting SVC model are better than those generated by the mean SVC model.

**References**

[1] Feng Shicon. Research on automatic Classification of Chinese Web pages and its Application in search engines. Ph. D. thesis, Peking University, 2003.
[2] Wang Tao. Research on automatic text Classification. Research on Library Science, July 12, 2007.

[3] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781, 2013.

[4] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffret Dean. Distributed Representations of Words and Phrases and their Compositionality. arXiv:1310.4546, 2013.