# Takeaway Comments Sentiment Analysis Based on BERT

**Jinlin Zhou**[1], **Haixin Song**[1], **Wendong Wang**[1], **Yao Niu**[1], **Wenhao Rao**[1]
[1]Department of Artificial Intelligence, Xiamen University
jnlnzhou@gmail.com

## Abstract

Analyzing the sentiment of the takeaway review information plays an important role in the improvement of the products by the merchants and the selection of the products by the users. However, Chinese takeaway reviews not only have the characteristics of short corpus and divergent characteristics, but also include the problem of polysemy. In response to these problems, this paper proposes sentiment analysis based on the pre-training model BERT. First, perform Chinese word segmentation on the data set to filter out meaningless data. Then fine-tune the BERT, extract the word vector, and embed the real emotional semantics into the model through the dynamic adjustment of the word vector by BERT according to the context. In order to prevent the model from overfitting, we extract part of the data and use maskelm, add dropout into model, try to decay the learning rate during the training process, and stop the training early. Finally, we compare our model with the classical text classification models TextCNN and TextRNN. Experiments show that algorithm model based on BERT is more effective than the traditional model in terms of short text sentiment analysis.

## Introduction

With the development of the Internet and the accelerating pace of social life, takeaway, a business model that integrates offline products and offline services, is favored by more and more customers. The Report on China's Food Delivery Industry Development in 2019 and the First Half of 2020. released by Meituan Research Institute in 2020 shows that the scale of China's food delivery industry in 2019 was 653.6 billion yuan, an increase of 39.3% compared to 2018; the scale of consumers was about 460 million People, an increase of 12.7% compared to the end of 2018. At the end of 2020, through 7,220 effective consumer surveys, it is concluded that user reviews are becoming more important, with 58.8% of consumers choosing word-of-mouth reviews. Takeaway reviews refer to the methods used by consumers to quantitatively evaluate the products or services provided by takeaway merchants. They are generally divided into positive reviews, neutral reviews, and negative reviews. Simply counting the number of emotional words cannot represent the user's emotional polarity. For example, "It's delicious, delivery is too slow." Does this mean good reviews or bad reviews? It is difficult to make effective judgments, which requires digging into the text. Accurate scores of takeaway merchants can not only help consumers quickly select qualified merchants, but also provide a channel for merchants to improve their products.

Sentiment analysis began to work in the 1990s. The traditional methods of text emotion classification are based on emotion dictionary and machine learning. An HMM model that incorporates vocabulary (Jin and Ho 2009). Firstly, different words are established, and sent to HMM for unified training. A dictionary-based method that can extract the corresponding emotional polarity for both explicit and implicit aspects (Ding, Liu, and Yu 2008). Further optimized the dictionary-based method, and achieved better performance by linking the kernel recognition aspect of the tree with the viewpoint (Nguyen and Shirai 2015). A method was proposed based on emotion dictionary to solve the problem of sentiment analysis of Chinese text (Zhang et al. 2017). The method proposed based on extended emotion dictionary is feasible and accurate for sentiment recognition of comment text (Xu et al. 2019). Using multi domain labeled dataset trained naive Bayesian bootstrapping multiple classifiers (Gamon and Aue 2005). Using naive Bayesian algorithm to obtain 80.48% classification accuracy (Tama, Sibaroni, and Adiwijaya 2019). However, the classification based on emotion dictionary relies too much on the constructed emotion dictionary, which is not universal. The method based on machine learning usually relies on complex feature process, and the cost of manual annotation is high.

## Related Work

In 2006, the concept of deep learning was proposed, that is, using deep learning networks to build a high-quality language model to deal with natural language problems (Hinton and Salakhutdinov 2006). With the advent of neural networks, researchers began to use deep learning extensively for sentiment analysis. The progress of sentiment analysis is advancing rapidly over time. Convolutional neural network CNN proposed for Weibo sentiment analysis in the literature. The author uses several comments to expand a Weibo to a combination of multiple Weibo corpus to solve the problem of short and sparse Weibo (Sun et al. 2015). CNN has three characteristics: local receptive field, weight sharing and down-sampling. It has great advantages in capturing local feature information of text. But for some information that needs to be combined with context, RNN is needed to deal with sequence problems.

Most of the current work is to combine CNN and RNN. Based on recurrent neural network RNN, result for microblog sentiment analysis is decent. A DCNN network model is proposed, which uses a dynamic pooling method that can handle variable length input (Kalchbrenner, Grefenstette, and Blunsom 2014). The two-way LSTM network can simultaneously capture the temporal relationship between words and obtain inter-words' context (Augenstein et al. 2016). After the CNN network, RNN is introduced to construct a hybrid model, which proves that the combination of multiple neural network models can better complete text sentiment analysis (Hassan and Mahmood 2017). But it still cannot solve the redundancy of the text.

In recent years, more and more researchers have added attention mechanisms to natural language processing to achieve automatic text alignment and improve text recognition accuracy. ATAE-LSTM model based on the attention mechanism (Wang et al. 2016). After the model encodes the sentence and the given aspect words with LSTM, the attention mechanism is used to process the hidden layer output, and the obtained attention. The vector and the aspect word vector are spliced to obtain the emotional polarity expression of the aspect word. Multi-head self-attention (MA) method can construct a Transformer model (Vaswani et al. 2017). In particular, the BERT model proposed by Google in 2018, as a substitute for Word2Vec, has greatly refreshed its accuracy in 11 directions in the NLP field (Devlin et al. 2018). It can be said that it is the best breakthrough technology of self-residual network in recent years with the following particularities:

- Use Transformer as the main framework of the algorithm, which can more thoroughly capture the two-way relationship in the sentence.
- The use of more powerful machines to train larger-scale data makes the results of BERT reach a whole new level.

The essence of BERT is to learn a good feature representation for words by running a self-supervised learning method on the basis of massive corpus. The model can be fine-tuned or fixed according to the task as a feature extractor.

## Application Framework

We used TextCNN, TextRNN and BERT to train the sentiment analysis of takeaway reviews. In the text classification task, CNN can be used to extract key information similar to n-gram in sentences.

Although TextCNN can perform well in many tasks, one of the biggest problems of CNN is to fix the filter size field of vision. On the one hand, it is unable to model longer sequence information. On the other hand, the adjustment of filter size parameters is very complicated. The essence of CNN is to express text features, and recurrent neural network is more commonly used in natural language processing, which can better express context information. In the task of text classification, bi-directional RNN (actually using bidirectional LSTM or GRU) can be understood as capturing variable length and bidirectional n-gram information.

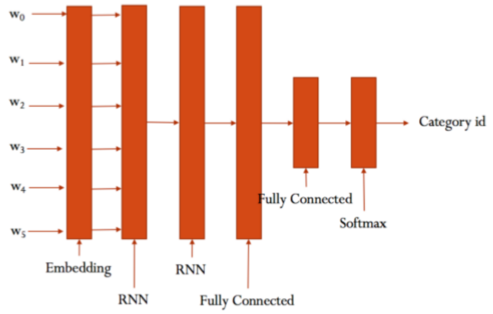


Figure 1: The architecture of the TextRNN.

BERT used Transformer, which is more efficient and can capture longer distance dependencies than RNN. Compared with the previous pre training model, it captures the real bidirectional context information. Next, I'll describe in detail the models used.

### TextRNN

In some natural language processing tasks, when processing sequences, we usually use the recurrent neural network RNN, especially its variants, such as LSTM (more commonly used), GRU. Of course, we can also apply RNN to text classification tasks.

The overall architecture of this model is shown in Figure 1. Generally, forward / reverse lstms are hidden in the last time step, and then stitched together to make a multi classification after passing through a softmax layer (the output layer uses softmax activation function); or take the hidden state of forward / reverse LSTM on each time step to splice the two hidden states on each time step, and then splice all the time steps After that, the hidden state takes the mean value, and then passes through a softmax layer (the output layer uses the softmax activation function) to perform a multi classification (the sigmoid activation function is used for the second classification).

### TextCNN

TextCNN model mainly makes one-dimensional convolution layer and time series most pooling layer. Suppose that the sequence of input word is composed of n words, and the word vector table of d dimension of each word is. Then the width of the input sample is n, the height is 1, and the number of input channels is d. The calculation of TextCNN can be divided into the following steps:

- We define several convolution kernels and make these convolution kernels do convolution computation separately. Convolution kernels with different widths may capture the correlation of different numbers of adjacent words.
- Max-over-time pooling all the output channels, and then the pooled output values of these channels are linked as vectors.

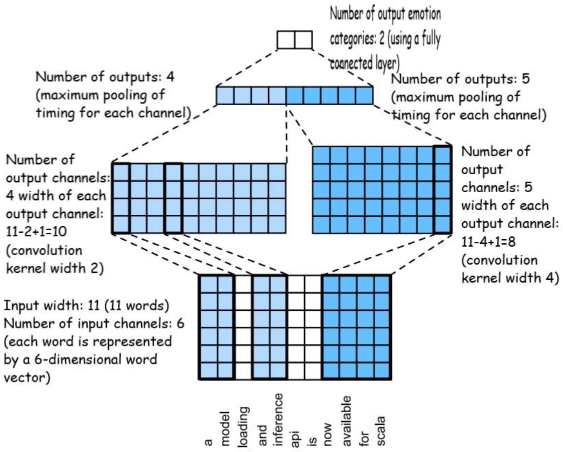

Number of output emotion categories: 2 (using a fully connected layer)
Number of outputs: 4 (maximum pooling of timing for each channel)
Number of outputs: 5 (maximum pooling of timing for each channel)
Number of output channels: 4 width of each output channel: 11-2+1=10 (convolution kernel width 2)
Number of output channels: 5 width of each output channel: 11-4+1=8 (convolution kernel width 4)
Input width: 11 (11 words) Number of input channels: 6 (each word is represented by a 6-dimensional word vector)
a
model
loading
and
inference
api
is
now
available
for
scala

Figure 2: The training process of the TextCNN.

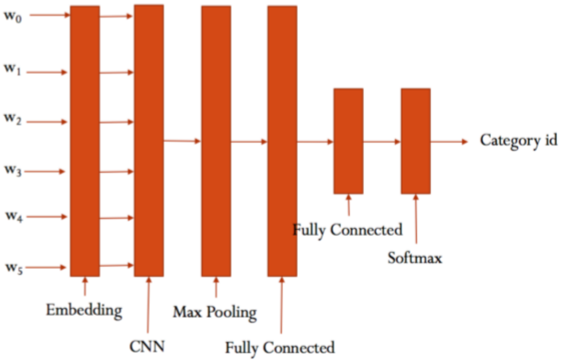


Figure 3: The architecture of the TextCNN.

- Through the full link layer, the linked vector is transformed into the output of each category. This step enables the discard layer to cope with over fitting.

Figure 2 and Figure 3 show the training process and architecture of TextCNN. The input here is 11 word sentences, each of which is represented by a 6-dimensional word vector. Therefore, the width of the input sequence is 11 and the number of input channels is 6. Given two convolution kernels, the kernel width is 2 and 4, and the number of output channels is 4 and 5. Therefore, after one-dimensional convolution calculation, the width of the four output channels is 11 - 2 + 1 = 10, while the width of the other five channels is 11 - 4 + 1 = 8. Although the width of each channel is different, we can still pool the output of 9 channels into 9-dimensional vectors. Finally, full connection is used to transform the 9-dimensional vector into 2-D output, that is, the prediction of positive emotion and negative emotion.

## BERT

The full name of BERT is bidirectional encoder representation from transformers, that is, the encoder of bidirectional transformer, because the decoder can not obtain the information to be predicted. The main innovation of the model

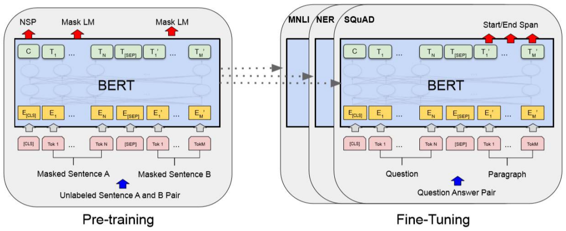


Figure 4: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating questions/answers).

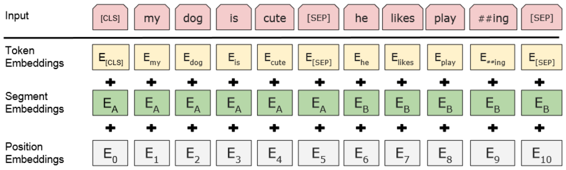
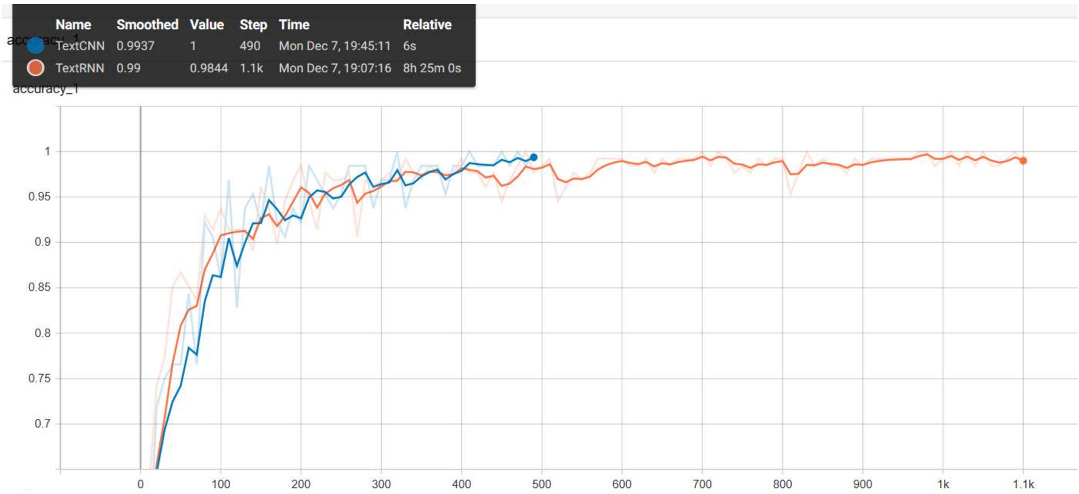


Figure 5: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

is pre train method, which uses masked LM and next sentence prediction to capture the representation of words and sentences respectively. Transformer is an encoder decoder structure(Vaswani et al. 2017). The model can calculate the attention relationship between input and output through three parameters of Q, K and V (query, key and value).

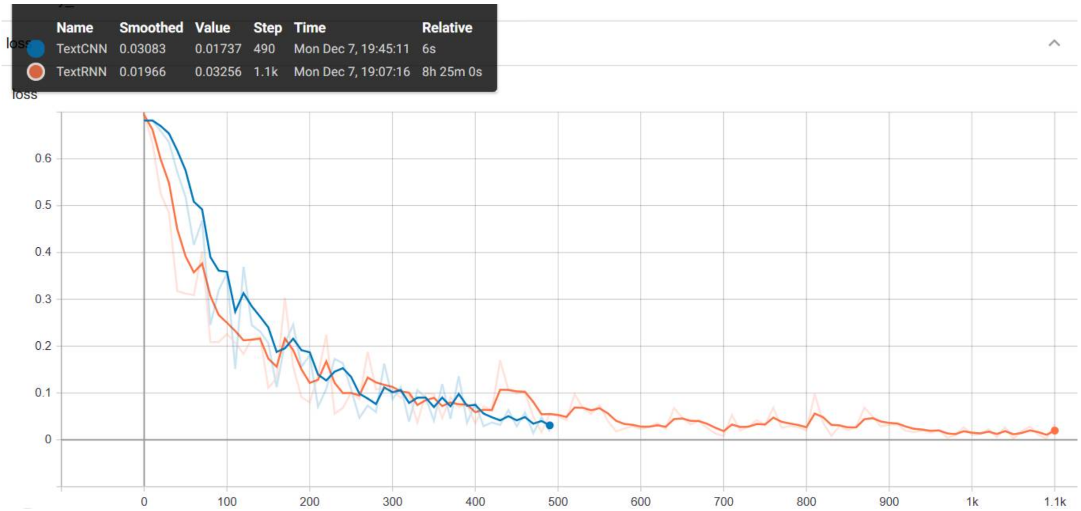
In pre training, BERT adopts MLM (masked language model) and NSP (next sentence prediction) methods to train the context between texts. Among them, MLM is a random mask of 15% of the words, NSP is to determine whether the next sentence is really the next sentence of the current sentence. Through these two methods, the attention relationship between texts is strengthened.

The structure of BERT is shown in Figure 4 and Figure 5. As shown in Figure 5, when inputting in pairs, the word is first converted into a token through wordpiece and other methods. In Chinese, each token is a single word. Next, we get an embedding of each sentence according to whether there is a context between the two sentences. Finally, because the attention mechanism will lose the position relationship before and after the text, a location code is added. By adding the above three embedding methods, the vector to be input into the model for training is obtained. In Figure 4, the blue area in the middle is a hidden layer composed of the encoder in the transformer. It is necessary to train and learn these parameters to obtain the corresponding output. After completing the pre training, we only need to change the output into corresponding tasks, such as emotion classification, question answering system, and then fine tune the



| | Name | Smoothed | Value | Step | Time | Relative |
|---|---|---|---|---|---|---|
| ● | TextCNN | 0.9937 | 1 | 490 | Mon Dec 7, 19:45:11 | 6s |
| ● | TextRNN | 0.99 | 0.9844 | 1.1k | Mon Dec 7, 19:07:16 | 8h 25m 0s |

Figure 6: The accuracy of TextRNN and TextCNN



| | Name | Smoothed | Value | Step | Time | Relative |
|---|---|---|---|---|---|---|
| ● | TextCNN | 0.03083 | 0.01737 | 490 | Mon Dec 7, 19:45:11 | 6s |
| ● | TextRNN | 0.01966 | 0.03256 | 1.1k | Mon Dec 7, 19:07:16 | 8h 25m 0s |

Figure 7: The accuracy of TextRNN and TextCNN

parameters according to the pre training.

Specifically, the calculation process of Q, K and V is as follows. Taking NLP task as an example, suppose that there are only four inputs ($x_1$, $x_2$, $x_3$, $x_4$), $a_i$ = $W_{xi}$. For $a_i$, multiply the weight matrix $W_q$, $W_k$, $W_v$ to obtain $q_i$, $k_i$, $v_i$. Next, for each location, take $a_1$ as an example, query the k of other locations separately, do the multiplication operation (vector dot product), if the $a_3$ is the most similar to $a_1$, then $a_{1,3}$ will get the maximum result. In this way, the attention relationship between each position and $a_1$ can be reflected by $b_1$. By analogy, $b_i$ reflects the representation of attention relationship between $a_2$ and other positions. After getting b, the decoder can use the different attention relations to obtain the corresponding output vector.

## Experiments

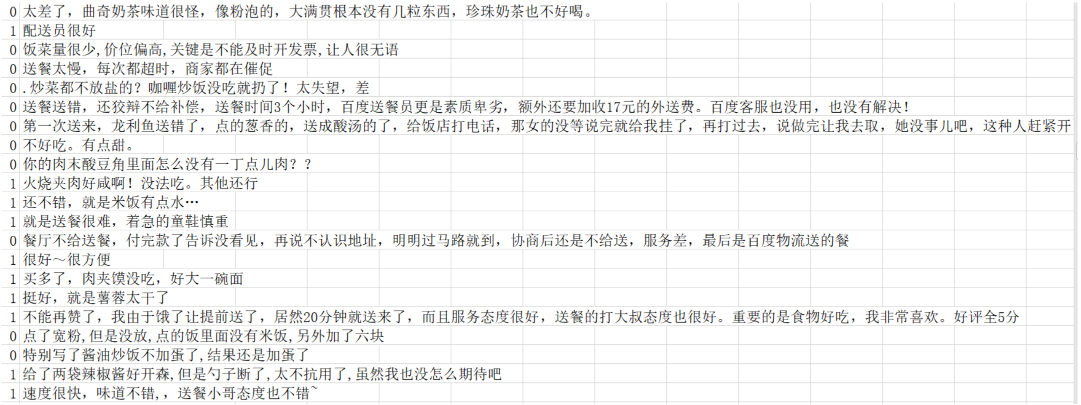
We perform experiments to evaluate TextCNN, TextRNN and BERT. We first discuss the dataset and model settings used in the experiments. We then compare and analyze the performance of these models with each other. Figure 6 and Figure 7 show the accuracy and loss of TextRNN and TextCNN.

### Dataset

We evaluate these models on the waimai_10k dataset. The waimai_10k is user reviews collected by a takeaway platform. A total of 8000 comments (4000 positive and negative comments) were extracted from the takeaway dataset. After scrambling, they were divided into training set 3200 (1600 for positive and negative), 3200 for verification set (1600 for positive and negative), and 1600 for test set (800 for positive and negative).

### Training Tricks

Due to BERT training is not good enough, we added some tricks. As shown follow:



| | |
|---|---|
| 0 | 太差了，曲奇奶茶味道很怪，像粉泡的，大满贯根本没有几粒东西，珍珠奶茶也不好喝。 |
| 1 | 配送员很好 |
| 0 | 饭菜量很少,价位偏高,关键是不能及时开发票,让人很无语 |
| 0 | 送餐太慢，每次都超时，商家都在催促 |
| 0 | .炒菜都不放盐的？咖喱炒饭没吃就扔了！太失望，差 |
| 0 | 送餐送错，还狡辩不给补偿，送餐时间3个小时，百度送餐员更是素质卑劣，额外还要加收17元的外送费。百度客服也没用，也没有解决！ |
| 0 | 第一次送来，龙利鱼送错了，点的葱香的，送成酸汤的了，给饭店打电话，那女的没等说完就给我挂了，再打过去，说做完让我去取，她没事儿吧，这种人赶紧开 |
| 0 | 不好吃。有点甜。 |
| 0 | 你的肉末酸豆角里面怎么没有一丁点儿肉？？ |
| 1 | 火烧夹肉好咸啊！没法吃。其他还行 |
| 1 | 还不错，就是米饭有点水… |
| 1 | 就是送餐很难，着急的童鞋慎重 |
| 0 | 餐厅不给送餐，付完款了告诉没看见，再说不认识地址，明明过马路就到，协商后还是不给送，服务差，最后是百度物流送的餐 |
| 1 | 很好～很方便 |
| 1 | 买多了，肉夹馍没吃，好大一碗面 |
| 1 | 挺好，就是薯蓉太干了 |
| 1 | 不能再赞了，我由于饿了让提前送了，居然20分钟就送来了，而且服务态度很好，送餐的打大叔态度也很好。重要的是食物好吃，我非常喜欢。好评全5分 |
| 0 | 点了宽粉,但是没放,点的饭里面没有米饭,另外加了六块 |
| 0 | 特别写了酱油炒饭不加蛋了,结果还是加蛋了 |
| 1 | 给了两袋辣椒酱好开森,但是勺子断了,太不抗用了,虽然我也没怎么期待吧 |
| 1 | 速度很快，味道不错,，送餐小哥态度也不错˜ |

Figure 8: The output of sentiment analysis training

- In this model, we separate the words of maskelm into two parts: one is the word of maskelm; the other is the word of maskelm.
- Many sentences contain numbers. Obviously, in masked LM, it is unrealistic to let the model accurately predict the data. Therefore, we replace the numbers (including integers and decimals) in the original text with a special token, so that the model can only predict that this place should be some numbers.

### Prevent Over Fitting

Due to the large parameters of the BERT model, severe over fitting and poor generalization ability may occur in training. Therefore, we will deal with this situation:

- Learning rate decay. We divide the data set into training and test sets. Our training method is to train each group with training set. The measurement standard of model performance is AUC. The measurement standard of AUC is very easy to use for two categories. Here, we don't talk about it because of the time relationship. After the current epoch training, we use the test set to measure the current training results and record the AUC of the current epoch, If the current AUC is not improved compared with the previous epoch, the learning rate will be reduced.
- Early stopping. If there is no obvious improvement within 100 epochs after the learning rate decay, the training will be terminated in advance.
- We also use the method of weight decay. Because our training data is very small, we have never seen some combination models of words and words or sentence structure when we use the model to infer. If the value of parameters in the model is large, the model will respond to the sentence too much when encountering some special sentences or words, In fact, we want the model to be more stable, so we add L2 normalization. The purpose of L2 regularity is to prevent the value of the parameter from becoming too large or too small.
- Dropout. Dropout is set to 0.4, because there are too many parameters in the model, so 40% of the parameters can be disabled during training to prevent over fitting.

| Model | F1 | Acc |
|---|---|---|
| TextCNN | 0.85632 | 0.85030 |
| TextRNN | 0.86964 | 0.84521 |
| BERT | 0.97558 | 0.92423 |

Table 1: Comparison of different models on the waimai_10k dataset

### Results

Table 1 shows the results of different models on the waimai_10k dataset. From the results, we can see that the BERT remarkably outperforms the TextRNN and TextCNN.

Figure 8 shows the output of emotion analysis training of the model. We can see that the result of the training is not bad.

From the results, it can be seen that the training effect of BERT is the best, but the training time is the longest. The effects of TextCNN and TextRNN are very close, but TextCNN is easier to train.

## Conclusion

In this paper, we use three models: TextCNN, TextRNN and BERT. We train the three models for sentiment analysis of takeaway reviews. The structure of TextRNN is very flexible and can be changed arbitrarily. For example, replacing GRU unit with LSTM unit, changing bidirectional to unidirectional, adding dropout or batch normalization and stacking one more layer. The effect of TextRNN on text classification task is very good, which is close to TextCNN, but the training speed of RNN is relatively slow, and generally two layers are enough. Through comparative analysis, it further highlights the advantages of BERT. In the future, we will try to add softmax after BERT to improve the training effect.

# References

Augenstein, I.; Rocktäschel, T.; Vlachos, A.; and Bontcheva, K. 2016. Stance detection with bidirectional conditional encoding. *arXiv preprint arXiv:1606.05464* .

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* .

Ding, X.; Liu, B.; and Yu, P. S. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the International Conference on Web Search and Web Data Mining, WSDM 2008, Palo Alto, California, USA, February 11-12, 2008*.

Gamon, M.; and Aue, A. 2005. Automatic identification of sentiment vocabulary: exploiting low association with known sentiment terms. In *the ACL Workshop*.

Hassan, A.; and Mahmood, A. 2017. Efficient deep learning model for text classification based on recurrent and convolutional layers. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 1108–1113. IEEE.

Hinton, G. E.; and Salakhutdinov, R. R. 2006. Reducing the dimensionality of data with neural networks. *science* 313(5786): 504–507.

Jin, W.; and Ho, H. H. 2009. *A novel lexicalized HMM-based learning framework for web opinion mining NOTE FROM ACM: A Joint ACM Conference Committee has determined that the authors of this article violated ACM's publication policy on simultaneous submissions. Therefore ACM has shut of.*

Kalchbrenner, N.; Grefenstette, E.; and Blunsom, P. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188* .

Nguyen, T. H.; and Shirai, K. 2015. Aspect-Based Sentiment Analysis Using Tree Kernel Based Relation Extraction. In *International Conference on Intelligent Text Processing and Computational Linguistics*.

Sun, X.; Gao, F.; Li, C.; and Ren, F. 2015. Chinese microblog sentiment classification based on convolution neural network with content extension method. In *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*.

Tai, K. S.; Socher, R.; and Manning, C. D. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075* .

Tama, V. O.; Sibaroni, Y.; and Adiwijaya. 2019. Labeling Analysis in the Classification of Product Review Sentiments by using Multinomial Naive Bayes Algorithm. *Journal of Physics: Conference Series* .

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, Y.; Huang, M.; Zhu, X.; and Zhao, L. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, 606–615.

Xu, G.; Yu, Z.; Yao, H.; Li, F.; Meng, Y.; and Wu, X. 2019. Chinese Text Sentiment Analysis Based on Extended Sentiment Dictionary. *IEEE Access* 7: 43749–43762.

Zhang, S.; Wei, Z.; Yin, W.; and Tao, L. 2017. Sentiment analysis of Chinese micro-blog text based on extended sentiment dictionary. *Future Generation Computer Systems* 81(APR.): 395–403.