WUST at the NTCIR-14 FinNum Task

Wei Wang, Maofu Liu, Zhenlian Zhang

School of Computer Science and Technology, Wuhan University of Science and Technology,
Wuhan 430065, China
liumaofu@wust.edu.cn

Abstract. This paper describes our system in the FinNum task at NTCIR-14, and the FinNum task is dedicated to identifying the category of a numeral from the financial social media data, i.e. tweet, for fine-grained numeral understanding. In NTCIR-14, the FinNum task contains two subtasks, the first one, named Subtask1 in this paper, is to classify a numeral into seven categories, i.e. Monetary, Percentage, Option, Indicator, Temporal, Quantity, and Product/Version Number, and the second one, named Subtask2 in this paper, extends the Subtask1 to the subcategory level, which classifies financial numerals into seventeen classes. In our system, we first complete the Subtask1, and then, on the basis of the seven categories, we separately classify numerals into corresponding subcategories according to the category. Our submitted system constructs the classification model based on Support Vector Machines (SVM) to identify the categories of numerals. In additional experiments, we adopt the Bidirectional Encoder Representations from Transformers (BERT) model to act as a multi-classifier, and the experimental results show that the BERT model is superior to the SVM model.

Keywords: Financial Numeral Classification; SVM; BERT

Team Name. WUST

Subtasks. Fine-Grained Numeral Understanding in Financial Tweet (FinNum)

1 Introduction

In the financial field, numeral is a crucial part of financial documents. In order to understand the detail of opinions in the financial documents, we should not only analyze the text but also need to assay the numeric information in depth. Due to the informal writing style, analyzing social media data is more challenging than analyzing news and official documents.

In the FinNum task, numerals are the objects of classification, and the context information of numerals in the document has a great influence on category identification. Our team has carried out the FinNum task as the text classification problem. The Subtask1 is one seven-classification task, and the subtask2 is one seventeen-classification task. All categories and subcategories in the FinNum task have been shown in Table 1. Our system uses the training set to complete Subtask1 at first, ensuring the first-level main category accuracy. In Subtask2, the training set is divided into seven parts corresponding to the seven first-level categories, and then confirms the seventeen second-level subcategories corresponding to each data by the first-level category obtained by subtask1. Fig.1 illustrates the classification process in detail.

Table 1. All categories and subcategories in the FinNum task.

Category	Subcategory	Category	Subcategory
Monetary	money	Percentage	relative
	quote		absolute
	change	Option	exercise price
	buy price		maturity date
	sell price	Indicator	indicator
	forecast	Temporal	date
	stop loss		time
	support or resistance	Quantity	quantity
		Product	product

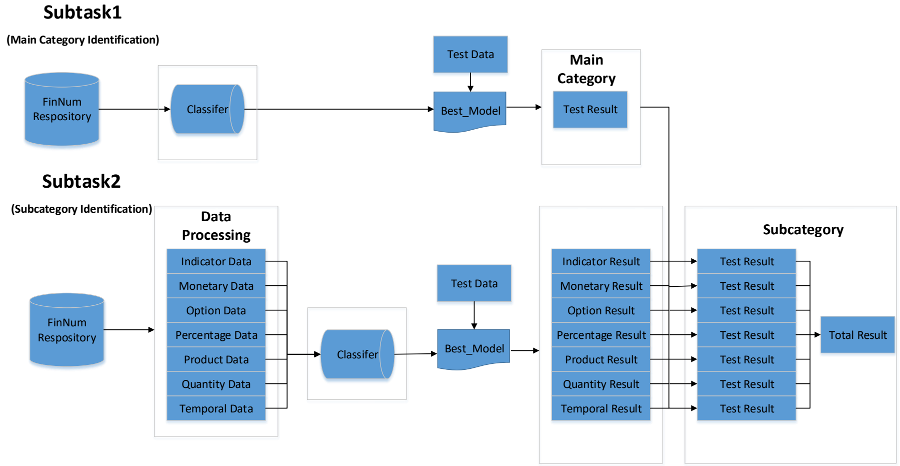

Fig. 1. The classification process.

For the small-scale and uneven training set, the traditional SVM [1,2] model would be more advantageous [3], so our submitted system adopted the SVM model.

In the additional system, we modified the BERT [4] model for category identification and conducted multiple classifications through the BERT model, which significantly improved the experimental results.

The rest of this paper is organized as follows. Section 2 shows the related work of Numeral Classification. Section 3 describes the data preprocessing, text features extraction and two different classification models, i.e. SVM and BERT. Section 4 shows the official experimental results and some discussions about error cases. Finally, we draw some conclusions in Section 5.

2 Related work

Text classification [5,6] is a common problem in NLP. For text classification, the common methods include Naive Bayes (NB) [7], Convolutional Neural Networks (CNN) [8], Recurrent Neural Network (RNN) [9] and Long Short-Term Memory (LSTM) [10]. In view of large-scale datasets, Zhang et al [11] proposed a character-level convolutional neural networks for text classification, showing that CNN can be directly applied to the distributed or discrete embedding of words, without any knowledge on the syntactic or semantic structures of a language. Lai et al [12] attempted to use RNN for text classification when getting rid of human-designed features. They apply a recurrent structure to capture contextual information to learn word representations as far as possible. The experiments results showed that RNN model also had a good performance in text classification, particularly on document-level datasets.

There are not many specific studies on the classification of financial texts. In 2009, Schumaker and Chen [13] proposed the AZFinText system to predict the breaking financial news from 9,211 financial news articles. In their experiments, the SVM model was used to perform a binary classification in two predefined categories, i.e. stock price rise and drop, which proved SVM is more suitable for small datasets.

In FinNum task, the dataset is from Stocktwits [14], including 4,072 training sets, 789 validation sets, and 343 test sets. Because of the small scale of datasets, our team chose the SVM model to act as a classifier. In the additional experiment, we employed the BERT model to replace the SVM model as the classifier to solve the problem of the FinNum task.

3 System Description

Based on the analysis of the corpus, the task can be regarded as a classification problem, which mainly identifies the numeral category by tweet. Our system includes three main modules, i.e. data preprocessing, feature extraction and classifier. Fig. 2 illustrates our system overview in detail.

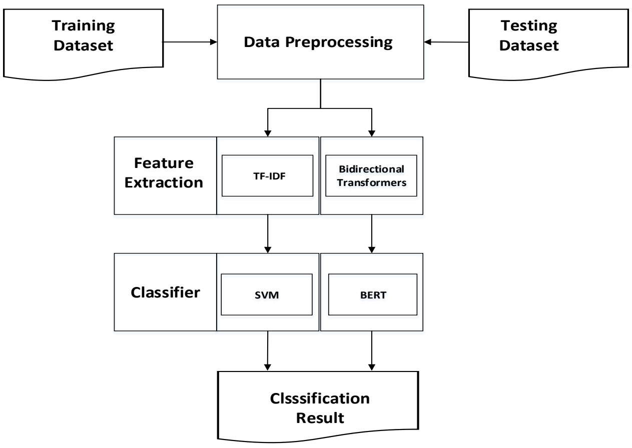

Fig. 2. System overview.

3.1 Data preprocessing

In the data preprocessing, we conduct statistics on 4,072 training data according to categories. Fig.3 demonstrates the distribution of data in 7 main categories: Indicator, Monetary, Option, Percentage, Product, Quantity, and Temporal. In Fig. 4, A shows the distribution of subcategories in Monetary, with a total of eight subcategories, and B stands for Option, C and D show Percentage and Temporal subcategories respectively. According to the data distribution, the data of main categories have the characteristic of the uneven distribution, and the Monetary and Temporal categories take a large proportion, accounting for 34% and 36% respectively, while the data of Indicator, Option, and Product take a small proportion, accounting for 2%, 3% and 2% respectively. In terms of subcategories, there are eight subcategories in Monetary, which are money, quote, change, buy price, sell price, forecast, stop loss, support or resistance. Percentage includes relative and absolute, and Options are divided into exercise price and maturity date. Temporal includes two subcategories, i.e. date and time. The subcategories of Indicators, Products and Quantity categories are the same with the main category.

We first do a simple processing on the original data. If a tweet contains multiple target num, we expand this training data into multiple training data, ensuring that one target num corresponds to one tweet and one category. As shown in Example 1 below.

Example 1:

Original data:
Target num: 22.5, 20
Tweet: $BZUN I sold 22.5 and 20 puts for income. Don't think I'll be assigned.
Expanded data:
Target num: 22.5
Tweet: $BZUN I sold 22.5 and 20 puts for income. Don't think I'll be assigned.
Target num: 20
Tweet: $BZUN I sold 22.5 and 20 puts for income. Don't think I'll be assigned.

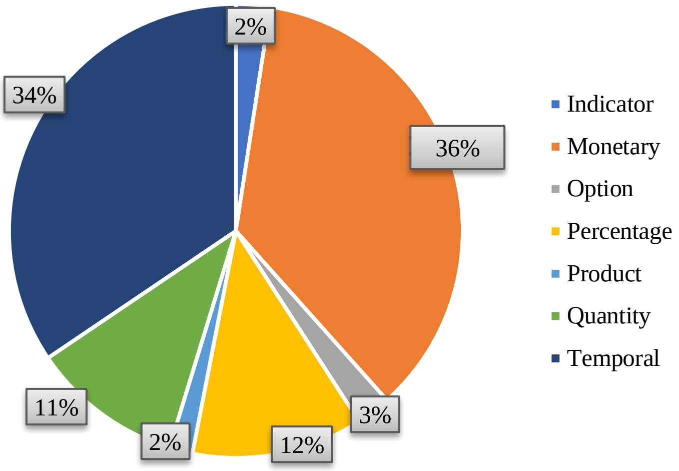

Fig. 3. The first-level category distribution.

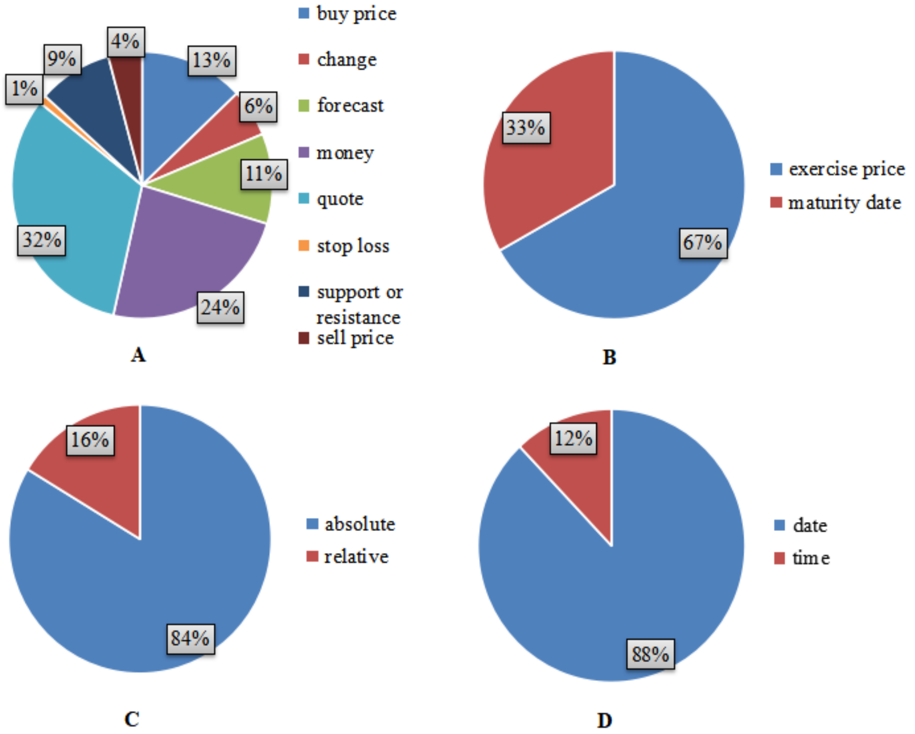

Fig. 4. The second-level subcategory distribution.

In Subtask1, due to the uneven data distribution, our team tried to expand the data of Indicator, Option and Product, divide tweets with the target num, and add the segmented data into the training set. As a result, the data of these three categories were tripled.

Example 2:
Original tweet:
Target num: 50
Tweet: $EGLT looking for that break above the 50 SMA. A little more volume will juice this baby much higher.
Split:
Target num: 50
Tweet: $EGLT looking for that break above the 50
Target num: 50
Tweet: 50 SMA. A little more volume will juice this baby much higher.

However, the experimental results show that the data expansion method reduced the accuracy of the validation set. The segmented tweet only retained part of the information and lost the contextual information when extracting textual features, leading to bad experimental results. Therefore, our team did not expand the experimental data.

In Subtask2, due to the large gap in the data distribution of Monetary,Temporal and Option categories, we expanded the data in the same way as Example 2.

3.2 Feature Extraction

All texts in the FinNum task are comments from Twitter, and our system splices target num and tweet as input, category as output. Before classifying with SVM, we extract textual features by Term Frequency-Inverse Documentation Frequency (TF-IDF) [15], and TF-IDF is a statistical method used to evaluate the importance of a word to a document set or one of the documents in the corpus. Term Frequency (TF) is the Frequency of a word in a document, as shown in Formula (1). Each distinct word corresponds to a feature, if a word appears more than once in the text, it may be more important. Fig. 5 shows an example feature vector for a particular tweet.

$$tf_{ij} = \frac{n_{ij}}{\sum_k n_{kj}} \tag{1}$$

Where the numerator represents the frequency of the word in the document, and the denominator represents the sum of the frequencies of all the words in the document.

Inverse Documentation Frequency (IDF) is the measure of word weights. The smaller the number of documents that contain the word, the larger the IDF will be, indicating that the word has a good ability of category discrimination, as shown in formula 2.

$$idf_i = log\frac{|D|}{1+|\{j: t_i \in d_j\}|} \tag{2}$$

The numerator represents the total number of tweets in the corpus, and the denominator represents the number of the word appearing in the document. If the word does not appear in the corpus, it will result in zero. Therefore, the denominator is generally plus one. The TF-IDF algorithm is simple in principle, fast in the calculation, and has strong universality for extracting keywords for FinNum task.

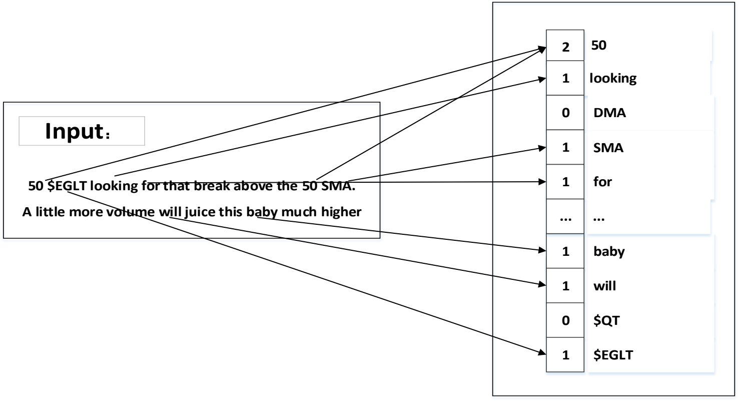
Input：
50 $EGLT looking for that break above the 50 SMA.
A little more volume will juice this baby much higher
2 50
1 looking
0 DMA
1 SMA
1 for
… …
1 baby
1 will
0 $QT
1 $EGLT

Fig. 5. Text representation in one feature vector.

3.3 SVM classifier

The SVM model has the characteristic of high precision and multidimensional data processing [16], and strong robustness, so the SVM model is favored in the study of classification problems. The SVM is usually used for binary classification problems, but the FinNum task is a multi-classification problem, so a multi-classification SVM needs to be constructed. For the problem of linearly separable samples, there are one-against-one and one-versus-rest.

For the problem of linear indivisibility, Kernel Function is needed to deal with it. Kernel functions solve the problem of linear indivisibility by mapping data to high order space. The advantage of kernel function is that it is calculated in low dimension in advance, and the classification effect is expressed in high dimension in essence, which avoids the complicated calculation in high dimension space. NLP classification is generally a nonlinear problem.

The classification performance of SVM is restricted by many factors, and kernel function selection is particularly important. There are four commonly used kernel functions, namely, Linear kernel function, polynomial kernel function, radial basis (RBF) kernel function, and Sigmoid kernel function. The selection of kernel function needs specific analysis. The Gaussian radial basis function is a kind of kernel function with strong locality. It can map a sample to a higher dimensional space. Both large samples and small samples have better performance, and its parameters are less than those of polynomial kernel function. So this paper chooses RBF kernel function, as shown in formula 3.

$$k(x, x_i = exp(-\gamma\|x_i - x_j\|^2)), \gamma > 0 \tag{3}$$

Where, γ denotes the width of the kernel function.

3.4 BERT classifier

The BERT is a method of pre-training language representations, meaning that it trains a universal language understanding model on a large-scale corpus, e.g. Wikipedia, and then use that model for downstream NLP tasks that we care about, e.g. question answering. The BERT outperforms previous methods because it is the first unsupervised, deeply bidirectional system for pre-training NLP [17].

The BERT uses a simple approach, in which a special classification embedding ([CLS]) inserted as the first token and a special token ([SEP]) added as the final token. It masks out 15% of the words in the input, runs the entire sequence through a deep bidirectional Transformer encoder, and then predict only the masked words.

Tweet:50 $EEM And there [MASK1] 50MA, let's [MASK2] $EDC.

Labels: [MASK1] = goes; [MASK2] = hold

Our system is based on the MRPC task which is Paraphrase Identification, feeding two sentences and then judging whether they represent the same meaning. The output is divided into two categories, i.e. yes and no. Our classification task needs only one input, rather than a pair of sentences, and the reading phrase can automatically identify and adjust the corresponding Sentence Embedding as shown in Fig. 6.

We adjusted the input and output, after adjusting, the code can automatically define the tag list according to the number of categories, which can adapt to a variety of multi-classification tasks.

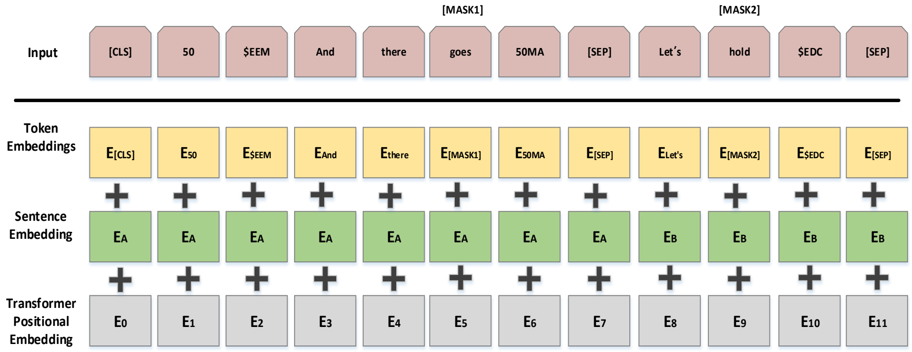

Fig. 6. Sentence embedding.

4 Experiments

4.1 Settings

In this paper, we use the data sets provided by NTCIR-14. This dataset contains about 6,840 training instances and 1,200 instances for testing. There are two values to evaluate the experimental results, namely Micro and Macro [18].

$$Precision = \frac{TP}{TP+FP} \tag{4}$$

$$Recall = \frac{TP}{TP+FN} \tag{5}$$

$$F1 - measure = \frac{2*Precision*Recall}{Precision+Recall} \quad (6)$$

Micro: Calculate metrics globally by counting the total true positives, false negatives and false positives.

Macro: Calculate metrics for each label, and find their unweighted mean. This does not take label imbalance into account.

Among them, the TP (True Positives) refers to the category where cases are truly classified into the positive class, FP (False Positives) refers to the category where cases are wrongly classified into the positive class, FN (False Negatives) refers to the category where cases are wrongly classified into the positive class.

4.2 Experimental results

We submitted one system result to NTCIR-14 for the FinNum task [19]. The official evaluation results are listed in Table 2 and Table 3.

Table 2. The official evaluation results of Subtask1.

Team	Task Name	Micro	Macro
Fortia1	Subtask1	0.9394	0.9005
DeepMRT	Subtask1	0.9187	0.8794
ASNLU	Subtask1	0.8972	0.8093
ZHAW	Subtask1	0.8645	0.7927
Fortia2	Subtask1	0.8988	0.7926
aiai	Subtask1	0.8645	0.7809
WUST	**Subtask1**	**0.7402**	**0.6371**
BRNIR	Subtask1	0.7427	0.6353
Stark	Subtask1	0.7801	0.6175

Table 3. The official evaluation results of Subtask2.

Team	Task Name	Micro	Macro
Fortia1	Subtask2	0.8717	0.8240
DeepMRT	Subtask2	0.8303	0.7790
aiai	Subtask2	0.8024	0.7411
ASNLU	Subtask2	0.7912	0.7251
Fortia2	Subtask2	0.7705	0.6886
ZHAW	Subtask2	0.7554	0.6644
Stark	Subtask2	0.6908	0.5683
WUST	**Subtask2**	**0.6088**	**0.5293**
BRNIR	Subtask2	0.6199	0.4714

From the experimental results, we can see that the SVM model has achieved good performance. However, there is still much room for improvement in the experimental results. Therefore, in the additional experiments, we selected the BERT model as a classifier. The experimental results are shown in Table 5.

Table 5. The additional evaluation results with the BERT model.

Team	Task Name	Micro	Macro
WUST	**Subtask1**	**0.9450**	**0.8862**
WUST	**Subtask2**	**0.8725**	**0.8307**

As can be seen from the experimental results, both Subtask1 and Subtask2 have greatly improved in the results with BERT classification model compared to SVM.

4.3 Error Analysis

In this subsection, according to the classification results of BERT and SVM models, our team analyzed several examples of classification errors and discussed the reasons for the errors. The classification result of BERT is better than that of SVM, and therefore, we first analyze some cases where BERT classification is correct but SVM classification is wrong, which are listed as follows.

(1) In Example 3. For the target num "85", the SVM classification result is Temporal, while the BERT classification result is correct and the category should be Monetary.

Example 3:
Target num: 85
Tweet: 2016 $QD post net income of 85M on total rev. of 212M In six months that ended June 30 of this year, rev. was 270M &amp;net income came at 143M

The target num is 85, the tweet is "2016 $QD post net income of 85M on total rev. of 212M In six months that ended June 30 of this year, rev. was 270M &amp;net income came at 143M". In this tweet, the word corresponding to "85" is "85M". The TF-IDF extracts feature do not recognize "85" as the keyword, resulting in SVM classification error. For BERT model, positional information is added in feature extraction, which can easily identify the correct category according to the contextual information.

(2) In Example 4. For the target num 4, the SVM classification result is Percentage, while the BERT classification result is correct and the category should be Product.

Example 4:
Target num: 4
Tweet: $AMD get in before mainstream gets the news. Nov NPD just released! Xbox and ps4 up 40% YOY.pro and X sell for big profits

The target num is "4", the tweet is "$AMD get in before mainstream gets the news. Nov NPD just released! Xbox and ps4 up 40% YOY.pro and X sell for big profits", In this tweet, The words "ps4" and "40%" both contain the number "4", while "%" is a strong future in Percentage category, which mislead the SVM to classify target num as Percentage for these reasons, However, the BERT model accurately identify the keyword corresponding to the target num as "ps4", and correctly classify the target num “4” as Product category according to the "Xbox" in the contextual information.

From the Example 3 and Example 4, it can be concluded that, compared with SVM model, because BERT model retains positional information when extracting textual features, it has more advantages in identifying a keyword in the tweet, and thus the classification effect is better. Despite the excellent classification results of BERT model, there are still some errors in classification, which are listed as follows.

(3) In Example 5. For the target num 60, the BERT classification result is Percentage, while the correct category should be Product.

Example 5:
Target num: 60
Tweet: $LODE just gonna keep watching all of these, and try to add more to lode, goal 60k shs.

The target num is 60, the tweet is “$LODE just gonna keep watching all of these, and try to add more to lode, goal 60k shs.”. When target num in the end, the system gets an error extracting the contextual information. as shown in Example 5, the content after the target num is just “shs”, which ultimately results in a misclassification to the Monetary category instead of the Quantity category.

(3) In Example 6, For the target num 0, the BERT classification result is Monetary, while the correct category should be Indicator. As shown in Example 6 below.

Example 6:
Target num: 0
Tweet: $IMUC Daily Chart - Stoch RSI Curling UP, CCI just crossed 0, MACD Curling up, Votex trending up , RSI below 50 - WTF you need?

In the process of classification, the amount of training data of some categories is too small, which leads to underfitting and also affects the final classification results [20]. In this task, the Indicator data is only 167, accounting for 2% of the total amount of training data, the system will under fitting Indicator category in training process, which cause the target num “0” to be incorrectly identified as the Monetary category.

5 Conclusions

In this paper, we construct the classification model based on SVM to identify categories of numerals in financial documents. In additional experiments, we employ BERT model to replace SVM for classification. The results of the experiments have been greatly improved. We will continue to improve our model for error analysis. Moreover, we have mainly taken statistical features into consideration, and we will extract and select more rules and semantic features for our system to improve the system accuracy.

6 Acknowledgments

The work presented in this paper is partially supported by the Major Projects of National Social Foundation of China under Grant No. 11&ZD189.

7 References

1. Suykens, J.A. and Vandewalle, J., 1999.: Least squares support vector machine classifiers. Neural processing letters, 9(3), pp.293-300.
2. Sun, A., Lim, E.P. and Ng, W.K., 2002, November.: Web classification using support vector machine. In Proceedings of the 4th international workshop on Web information and data management (pp. 96-99). ACM.
3. Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov.: 2016b. Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759.
4. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR, abs/1810.04805.
5. Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a.: Fasttext.zip: Compressing text classification models. arXiv preprint arXiv:1612.03651.
6. Moraes, R, J.F. Valiati, and W.P.G. Neto.: "Document-level sentiment classification: An empirical comparison between SVM and ANN". Expert Systems with Applications, 2013. 40(2): p. 621-633.
7. McCallum A, Nigam K.: A comparison of event models for Naive Bayes text classification[C].AI-98 Workshop on Learning for Text Categorization. 1998, 752(1): 41-48.
8. Kim Y.: Convolutional Neural Networks for Sentence Classification[J]. Eprint Arxiv, 2014.
9. Hüsken M, Stagge P.: Recurrent neural networks for time series classification[J]. Neurocomputing, 2003, 50(none):223-235.
10. Graves A.: Long Short-Term Memory[M]. Supervised Sequence Labelling with Recurrent Neural Networks. 2012.
11. Zhang X, Zhao J, LeCun Y.: Character-level convolutional networks for text classification[C] Advances in neural information processing systems. 2015: 649-657.
12. Lai S, Xu L, Liu K, et al.: Recurrent convolutional neural networks for text classification[C] Twenty-ninth AAAI conference on artificial intelligence. 2015.
13. Schumaker R P, Chen H.: Textual analysis of stock market prediction using breaking financial news: The AZFin text system[J]. ACM Transactions on Information Systems (TOIS), 2009, 27(2): 12.
14. Chen C C, Huang H H, Shiue Y T, et al.: Numeral Understanding in Financial Tweets for Fine-grained Crowd-based Forecasting[C]. 2018 IEEE/WIC/ACM International Conference on Web Intelligence. IEEE, 2018: 136-143.
15. Forman G.: BNS feature scaling:an improved representation over tf-idf for svm text classification[C]. Acm Conference on Information & Knowledge Management. 2008.
16. Mathur A, Foody G M.: Multiclass and Binary SVM Classification: Implications for Training and Classification Users[J]. IEEE Geoscience & Remote Sensing Letters, 2008, 5(2):241-245.
17. Devlin J, Chang M W, Lee K, et al.: Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

18. Yang Y.: An evaluation of statistical approaches to text categorization[J]. Information retrieval, 1999, 1(1-2): 69-90.
19. Chen C C, Huang H H, Takamura, et al.: Overview of the NTCIR-14 FinNum Task: Fine-Grained Numeral Understanding in Financial Social Media Data. Proceedings of the 14th NTCIR Conference on Evaluation of Information Access Technologies. 2019.
20. Van der Aalst W M P, Rubin V, Verbeek H M W, et al.: Process mining: a two-step approach to balance between underfitting and overfitting[J]. Software & Systems Modeling, 2010, 9(1): 87.