Text Classification Model Based on fastText

Tengjun Yao
The 36th Institute
China Electronics Technology Group Corporation
Jiaxing, China
yaotj@jec.com.cn

Zhengang Zhai
The 36th Institute
China Electronics Technology Group Corporation
Jiaxing, China
zhaizg@jec.com.cn

Bingtao Gao
The 36th Institute
China Electronics Technology Group Corporation
Jiaxing, China
gaobt@jec.com.cn

***Abstract*—Most text classification models based on traditional machine learning algorithms have problems such as curse of dimensionality and poor performance. In order to solve the above problems, this paper proposes a text classification model based on fastText. Our model explores the important information contained in the text through the feature engineering, and obtains the low-dimensional, continuous and high-quality text representation through the fastText algorithm. The experiment is based on Python to classify the text dataset of "user comment data emotional polarity judgment" in Baidu Dianshi platform. In the emotional polarity judgment task, the experimental results show that the precision, recall and F values of our model are superior to the model based on traditional machine learning algorithms and have excellent classification performance.**

Keywords—Machine learning, text classification, feature engineering, emotional polarity judgment

I. INTRODUCTION

With the rapid development and rapid popularization of Internet technology, electronic text information data has shown explosive growth. How to effectively mine the information required by users from massive text data is a major challenge in the field of information science and technology. As a key technology for processing massive text data, text classification can more conveniently locate the required resources and improve the efficiency of data utilization. This solves the problems caused by massive text information to a certain extent.

Commonly used text classification methods include the classification method based on the Vector Space Model (VSM) [1], Naïve Bayes (NB) [2][3], and k-NearestNeighbor , kNN) [4], LDA model (Latent Dirichlet Allocation) [5], Support Vector Machine (SVM) [6], neural network method [7], etc. Since most of these methods use co-occurrence matrix as text representation, this leads to the problem of dimensional explosion of text representation. High-dimensional representations often fail to capture the main semantics of the text, which makes the performance of text classification models based on these algorithms often poor. The fastText model [8] can represent text in a low-dimensional, continuous space, and can capture the main semantics of the input text. Therefore, this paper proposes a text classification model based on fastText. The model takes user review text data of the merchant as input, and automatically discovers the emotional polarity (positive, negative, neutral) in the user evaluation, which can help the merchant to self-monitor and improve its product quality and service level. At the same time, it can also effectively reduce the manual discrimination cost of the merchant.

II. RELATED WORK

A. Softmax Regression

Softmax regression is also called polynomial logistic regression. In softmax regression, the training set can be expressed as (1).

$$\left\{\left(x^{(1)},y^{(1)}\right),...,\left(x^{(m)},y^{(m)}\right)\right\},y^{(i)}\in\left\{1,2,...,K\right\} \tag{1}$$

Given an input x, the model outputs a K-dimensional vector, and each element value of the vector represents the probability that x belongs to the current category. Specifically, the model can be represented by $h_\theta\left(x\right)$, as shown in (2).

$$h_\theta(x)=\begin{bmatrix}P(y=1\mid x;\theta)\\P(y=2\mid x;\theta)\\...\\P(y=K\mid x;\theta)\end{bmatrix}=\frac{1}{\sum_{j=1}^{K}e^{\theta^{(j)T}x}}\begin{bmatrix}e^{\theta^{(1)T}x}\\e^{\theta^{(2)T}x}\\...\\e^{\theta^{(K)T}x}\end{bmatrix} \tag{2}$$

The corresponding loss function of the model is shown in (3).

$$J(\theta)=-\left[\sum_{i=1}^{m}\sum_{k=1}^{K}1\left\{y^{(i)}=k\right\}\log\frac{e^{\theta^{(k)T}x^{(i)}}}{\sum_{j=1}^{K}e^{\theta^{(j)T}x^{(i)}}}\right] \tag{3}$$

B. Fasttext

Neural networks perform well in various tasks in the field of natural language processing, but there are also significant problems, that is, the training of the model is time-consuming. In order to solve this problem, Facebook Research in 2016 open sourced fastText, a tool for obtaining word vectors and text classification. The model architecture of fastText is shown in Fig. 1.

Compared to the traditional bag-of-words model, the input layer of the fastText model not only takes the word representation corresponding to each word in the sentence as

input, but also uses the n-gram feature of the sentence as an additional feature to input. For example, for the sentence "I like play basketball", the input of the traditional bag-of-words model is "I", "like", "play", and "basketball", and fasText is the basis of the bag-of-words model. An additional n-gram feature is added to the above. If n = 2, the additional feature is the average value of the words corresponding to "I like", "like playing", and "playing basketball". With the introduction of n-gram features, fastText can obtain the word order information in a sentence to a certain degree, thereby obtaining a more accurate sentence representation, which is difficult to achieve with traditional bag-of-words models.

In the hidden layer, fastText averages the word representation and n-gram features from the input layer. In the output layer, fastText uses hierarchical softmax to predict the label of the input text. The use of hierarchical softmax also greatly reduces model training time.

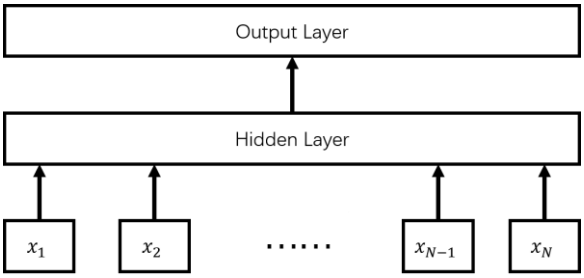

Fig. 1. The model architecture of fastText

III. Text Classification Model Based on fastText

The text classification model based on fastText mainly include modules for data preprocessing, feature extraction, training and evaluation.

A. Data Preprocessing

Because the corpus used in this paper is a Chinese corpus, all training corpora need to be tokenized first. After the word tokenization of the corpus is completed, the corpus is captured from the network platform and contains some meaningless garbled characters and expressions, such as 'hellip', '& hellip'. Therefore, in addition to removing general stop words, we also filters some meaningless characters from the Internet to improve the quality of the corpus.

B. Feature Extraction

In the previous chapter, we mentioned that fastText's text classification algorithm uses each word representation in the input text and n-gram features as input. The text classification model proposed in this paper is based on the fastText algorithm and adds field features based on the characteristics of the actual text corpus. An example of the text corpus used in this article is as follows.

33575	Financial Services	Very easy to use and very convenient	2

It can be seen that, in addition to the user's comment text and human-labeled emotional polarity accidents, the industry category to which the comment belongs is also expected (in the example, the comment comes from the financial services industry). In order to use as much information as possible in the corpus, we use the industry category to which each comment belongs as an independent feature at the input layer of the model, together with word representation and n-gram features, as the fastText-based Input for text classification algorithms. Since each industry may have its own unique terminology, the addition of industry category characteristics can theoretically effectively improve the performance of the classification model and further improve the accuracy and recall of the classification model.

C. Training

The training of text classification model based on fastText. The hyperparameters involved in the model are shown in Table I.

TABLE I. The hyperparameters involved in the model

Hyperparameter	description
Learning rate	A tuning parameter in an optimization algorithm that determines the step size at each iteration while moving toward a minimum of a loss function
Epoch	Set the algebra of model parameter iterations
Window size	Used to set the size of the context window
Bucket	Because there are many n-gram features, the hash method is used for mapping. Multiple n-grams may be mapped into the same bucket
Loss	Set the type of loss function, this paper uses the loss function of hierarchical softmax
Dim	The dimension of the word vector used to set the fastText parameter

The hyperparameters involved in the text classification model based on fastText are shown in Table II.

TABLE II. hyperparameters of the text classification model based on fasttext

Hyperparameter	Value
Learning rate	0.01
Epoch	300
Window size	7
Bucket	100000
Loss	hs
Dim	300

IV. Experiment

A. Experiment Environment

The experiments were performed on a notebook platform with an Intel (R) Core (TM) i5-8250u 1.60GHz, 8G memory, and an operating system of Windows 10 (64-bit home Chinese version). The algorithm is implemented by python 3.7. The main toolkits used are sklearn and fastText.

B. Data

The data used for the experiments in this article comes from the "User Comment Data Emotional Polarity Judgment" contest in Baidu Dianshi Data Contest(*https://dianshi.baidu.com/competition/18/rank*), which is user comment data on businesses in different industries. Each piece of data contains four fields, which are text id, type, comment content, and emotional polarity. Among them, the emotional polarity is cross-labeled by natural persons multiple times to ensure the reliability of data and scores. The detailed format of the experimental data is shown in Table III.

TABLE III. EXPERIMENTAL DATA FORMAT

Field name	Example
Text Id	9
Type	Food and catering
Comment content	It's great that companies today can do this
Emotional polarity	2

The user-commercial review data used in the experiments in this paper are mainly from five different industry areas, namely food and catering, travel accommodation, financial services, medical services and logistics express. The total amount of data is 82,025. We used Python to count the number and proportion of user reviews in each category in the experimental data. The distribution of the number of review categories in the experimental data is shown in Fig. 2 .

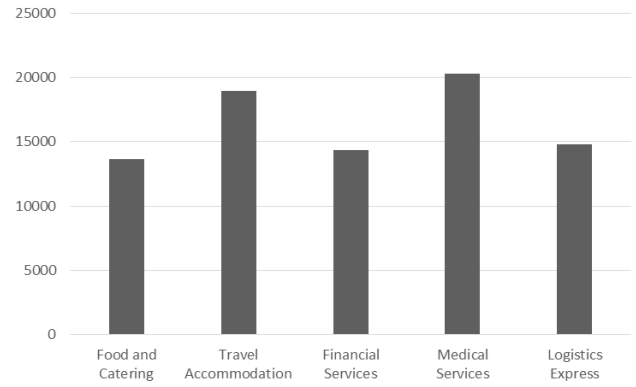

Fig. 2. The distribution of the number of review categories in the experimental data

There are three types of emotional polarity labels in the experimental data, including negative, neutral and positive, which are represented by 0, 1, 2 respectively. The number and proportion of comments corresponding to each label (polarity) in the data set are statistics, and the statistical results are shown in Fig. 3.

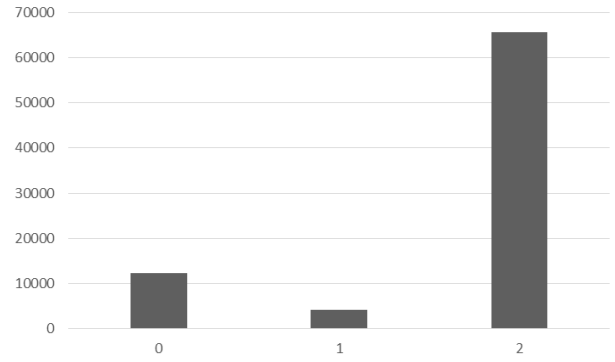

Fig. 3. Statistical results on experiment data

C. Experimental Results

In order to make a more objective and reasonable evaluation of the performance of the text classification model based on fastText, this paper applies support vector machine, k-nearest algorithm and Naive Bayes algorithm to text classification for comparison experiments..

During the experiment, the experiment of each model was repeated 10 independently. In each experiment, the data set was randomized, and divided into a training set and a test set according to a ratio of 4: 1. Then the model was trained through the training set, and the performance of the model was evaluated through the test set. The precision, recall and F value of each model are the average of 10 independent experiments. The experimental results are shown in Table IV.

TABLE IV. RESULTS

Model	Precision	Recall	F-value
SVM	0.9141	0.9248	0.9155
KNN	0.8812	0.6338	0.6712
NB	0.8636	0.8830	0.8525
Our model	**0.9275**	**0.9301**	**0.9286**

From the above experimental results, we can see that among the four text classification models, the F value calculated by the classification results of the k-neighbor algorithm is lower, and the overall performance of the model is significantly worse than the other three algorithms. The main reason may be that the number of positive samples is much more than the number of neutral and negative samples. At the same time, because the k-nearest algorithm model is relatively simple, a large amount of training data cannot be used well, and the performance of the model will also be reduced. Compared with the k-neighbors algorithm, the Naive Bayes algorithm has better recall than the k-neighbors algorithm, and the F value of the Naive Bayes algorithm model is also significantly higher than that of the k-neighbors algorithm. Support vector machine, as a classic classification model, has shown better performance than k-neighboring algorithm and Naive Bayes algorithm in user sentiment polarity classification task, and achieved a high F-value.

Among the four models, the text classification model based on fastText has achieved the best performance in the three indicators of accuracy, recall, and F value. This shows that compared with the traditional machine learning classification algorithm, the text classification model based on fastText can better capture and use the features related to emotional polarity in the input text, so as to make more accurate judgments on the emotional polarity of user evaluations. In the end, the text classification model based on fastText proposed in this paper won the tenth place in the "User Comment Data Emotion Polarity Judgment" contest in Baidu Point Data Competition (a total of 176 participating teams).

V. CONCLUSION

The text classification model based on fastText proposed in this paper, while making full use of the low-dimensional, continuous and high-quality sentence representation generated by the fastText, effectively mines the main information contained in the data. Through the combination of the two, the text classification model based on fastText shows superior performance. By comparing the proposed method with commonly used classification algorithms, it is proved that the text classification model based on fastText has the excellent performance.

However, text preprocessing and feature engineering also greatly affect the performance of the classifier. When tokenization is performed in the text preprocessing stage, part of the tokenization results fail to achieve the expected results, so the accuracy of the tokenization is also a direction for future research and exploration. In addition, although feature engineering has achieved good results, it has not explored the impact of different features on the final classification results. In the future, machine learning can be used to assign different weights to features in order to obtain

better text classification models.

REFERENCES

[1] Liu Shaohui, Dong Mingkai, Zhang Haijun, Li Rong, Shi Zhongzhi, "An Approach of Multi-hierarchy Text Classification Based on Vector Space Model," Journal of Chinese Information Processing, vol. 16, pp. 9-15,27, 2002.

[2] He Ming, Sun Jianjun, Cheng Ying, "Text Classification Based on Naive Bayes: A Review," Information Science, vol. 34, pp. 147-154, 2016.

[3] Di Peng, Duan Liguo, "New Naive Bayes Text Classification Algorithm," Journal of Data Acquisition & Processing, vol. 29, pp. 71-75, 2014.

[4] Zhang Ning, Jia Ziyan, Shi Zhongzhi, "Text Categorization with KNN Algorithm," Computer Engineering, vol. 31, pp. 171-172, 2005.

[5] Yao Quanzhu, Song Zhili, Peng Cheng, "Research on text categorization based on LDA," Computer Engineering and Applications, vol. 47, pp. 150-153, 2011.

[6] Fan Kangxin, "Research and design of network text sentiment classification system based on SVM," Computer Era, vol. 12, pp. 34-37, 2015

[7] Liu Tengfei, Yu Shuangyuan, Zhang Hongtao, Yin Hongfeng, "Recurrent Neural Networks and Convolutional Neural Networks for Text Classification," Computer Engineering & Software, vol. 39, pp. 64-69, 2018.

[8] Joulin A , Grave E , Bojanowski P, "Bag of Tricks for Efficient Text Classification," unpublished.