Short text classification based on word2vec and improved TDFIDF merge weighting

ZhenChen
Wuhan University of Science and Technology
Wuhan, 430065, China
Email: 294542248@qq.com

***Abstract*—With the development of information technology, the amount of information in contemporary society has proliferated exponentially. It contains not only the information we need, but also the redundant information that we do not need. Therefore, how to pass the massive data the algorithm processing, filtering out the information we do not need, has always been a hotspot in the field of information retrieval. TF-IDF is one of the statistical methods, but the traditional TF-IDF method simply highlights the importance of small frequency vocabulary, but it does not reflect the role of the keyword in the context; once encountered some very useful vocabulary, it will have a bad influence on the classification results. Therefore, we introduce the word2vec model and the improved TFIDF algorithm for combining weights. Experiments show that the model retrieval results are better than the traditional two models.**

Keywords- TFIDF; word2vec; Short text classification; Imformation retrieval

I. INTRODUCTION

With the rise of a large number of social media platforms, how to push customized information for user preferences is a hot research direction in the field of information retrieval; short text is a collection of information, generally refers to a length of no more than 160 characters. The text type is embodied in Weibo, literature search, hot comment and WeChat circle of friends. Due to the large number of social users, the short text has the characteristics of non-standard text fast update and few feature words. Therefore, the reasonable classification of short texts through certain statistical methods and computer technology has practical significance and great research value for mining user preferences, exploring public opinion and preference customization

In the traditional text document representation, the general methods include continuous bag model (CBOW), word frequency and inverse document frequency model (TFIDF) [1]. These methods simply verify the relationship between words and the entire document set, and are lost. The semantic information carried by the words themselves. In the research of text categorization, it is generally based on Vector Space Model [2]. This method has achieved good results when dealing with long texts [3], but under the characteristics of short text sparseness and irregularity, VSM classification effect Not satisfactory. In recent years, the short text classification by understanding the semantic information in the text has gradually become a research hotspot; BLEI et al. proposed an LDA (Latent Dirichlet Allocation) method for web short text classification [4], However, this method relies on a large amount of external corpus, and the training time takes a long time. Word2vec is a word vector training tool launched by Google in 2013. It uses a distributed vector to represent the text [5], and it is essentially a three layered neural network. By understanding the relationship between a word in the context, it eliminates the hassle of manually tagging large amounts of data. Compared with VSM, it solves the problem of high dimensional sparsity in traditional text representation models. However, it cannot measure the importance of the word in text. Therefore, this paper aims to improve the deficiencies of traditional TFIDF text classification methods, and proposes a classification model based on the characteristics of word2vec.

978-1-7281-3584-7/19/$31.00 ©2019 IEEE

II. METHODS

A. Model Introduction

TF-IDF is a commonly used data mining technology, in which TF (Term Frequency) represents the frequency of a given word in the text; IDF (Inverse Document Frequency) is a measure of the universality of a word. The essence of TF-IDF is actually the product of TF and IDF, which reflects the importance of a word for the entire document set. The traditional TF can be summarized by the formula (1):

$$tf_{ij} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (1)$$

The numerator is the number of times the keyword t appears in the document d_j, and the denominator represents the number of all keywords in the document d_j. The traditional IDF can be summarized by the formula (2):

$$idf_{i,j} = log\frac{|D|}{1+|D_{t_i}|} \quad (2)$$

|D| represents the total number of texts in a document set, $|D_{t_i}|$ indicates that a certain text contains the number of feature words t_i, and the denominator adds 1 to prevent the feature word from being absent in the document set and causing the denominator to be zero.

Word2vec is essentially a shallow neural network that maps each word into a vector to measure the degree of similarity between two words. It contains two training models, CBOW and skip-gram. Compared to the way TFIDF is characterized by keywords for text categorization, it captures semantic information between keyword contexts, but it is not suitable for

classification. We use the TFIDF method to form a complement with the word2vec merge weight, which is expected to improve the accuracy of text classification.

B. Traditional TF-IDF issues and improvements

The traditional TF-IDF illegible vocabulary will be judged as a key word for distinguishing in the TF. Such classification is not representative, So the improved formula is shown in (3):

$$tf_{ij}=\frac{N}{S(S\in D)} \tag{3}$$

Where, N is the number of times the keyword t appears in the text, and S is the number of all keywords in the text. When the text appears in a certain type of text, the tf_{ij} becomes smaller, thus reducing the noise generated by the uncommon words for text classification.

In the traditional TF-IDF, the IDF part only considers the relationship between the keyword and the number of text sets it appears, ignoring the distribution among different categories in a category, the improved formula is shown in (4):

$$idf_{ij} = \log\left(\frac{|D|*(t_i)_{max}}{1+|D_{t_i}|*(|D_{t_i}|-(t_i)_{max})}\right) \tag{4}$$

$(t_i)_{max}$ represents the maximum value of the number of texts of the i[th] feature word t in each category. When $|D_{t_i}|$ is fixed, the larger the $(t_i)_{max}$ is, the smaller the denominator is. The larger the idf_{ij} is, the more the keyword t is concentrated in this document set. Obviously, the improved formula can solve the above problem.

C. Word2vec training model selection

The CBOW model uses the surrounding vocabulary to predict the central vocabulary, while the skip-gram is the opposite; the CBOW method will be used as the central vocabulary after the training is completed. It can be seen that the CBOW method predicts the number of behaviors approximately equal to the total amount of words [6]. The skip-gram method uses the central word to predict the surrounding vocabulary each time. The time complexity is about n times that of CBOW (n is the window size). Skip-gram is more time-consuming but has better classification accuracy than CBOW [7] So we use the skip-gram model to train word vectors.

III. EXPERIMENT

A. Data processing

The data set used in this experiment is that THUC news is filtered and screened according to the historical data of Sina News RSS subscription channel between 2005 and 2011. After integration, 14 thousand categories of data are obtained, totaling 700,000 pieces are UTF- The 8 format documents are finance, lottery, real estate, stock, home, education, technology, society, fashion, politics, sports, constellation, games, entertainment. We randomly extracted 10,000 documents from each category, that is, a total of 140,000 documents. We take half of the categories from the training set and the other half as the verification set. In the experiment, the jieba word segmentation tool was used for word segmentation, and the stop words have been filtered, which can make the classification get better results.

B. Experiment process

- The jieba word segmentation tool generates a corpus training dictionary S, and a text collection n:

$$S=\{m_i|i \in N\} \tag{5}$$

Where, N is the word vector space dimension.

- We use Word2vec model to train corpus S, and sum the word vectors in corpus S to get the vector representation V_s of corpus S.

$$V_s=\sum_m w2v(m) \quad (m \in n) \tag{6}$$

w2v(m) represents the word vector of the word m.

- The weight of the word in word2vec is calculated according to the importance of the word by the improved TF-IDF method. After the weighted summation, we have:

$$W_{V_S}=\sum_m w2v(m) * weight(m) \tag{7}$$

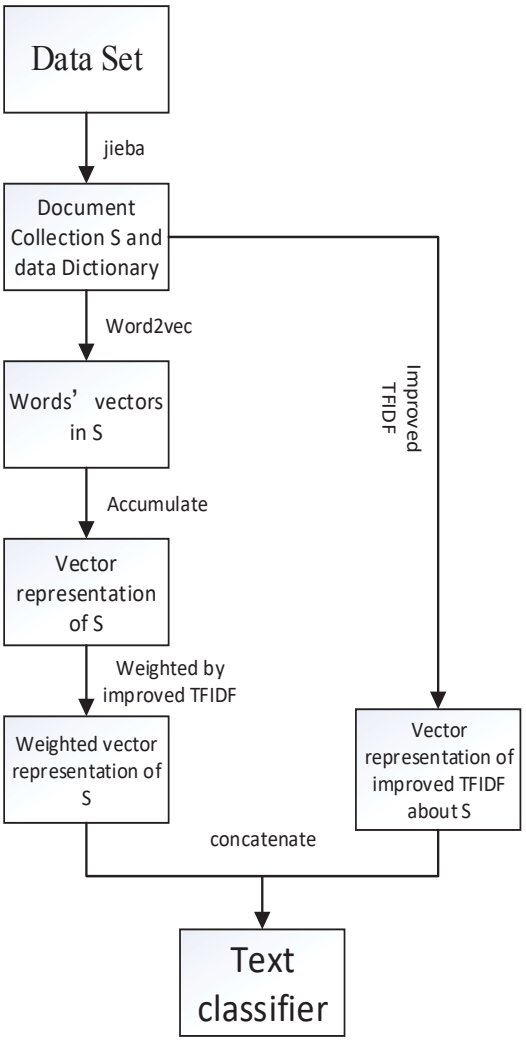
- The vector feature of the document is extracted by the improved TF-IDF method and vectorized to obtain TFIDF(n)
- The text is vectorized by combining the weighted word2vec and the improved TF-IDF by the concatenate method in Keras.

$$\text{Result} = \text{Concatenate}(W_{V_S}, TFIDF(n)) \tag{8}$$

- Use the LinearSVM method for classification.

C. Algorithm flow and experimental environment

The experimental platform is:
CPU: core i7-7700
GPU: Nvidia GTX 1060
RAM: 16Gb
Hard disk: 240GB
OS: Windows 10 1803 version
IDE: PyCharm Community
The algorithm flow is shown in (9):



(9)

IV. EXPERIMENTAL RESULT

In this experiment, two comparison models were used for comparison. The traditional TF-IDF model and the merging weighted TFIDF and word2vec model, the combination of categories based on the sample and prediction can be divided into four situation indicators as shown in the (10) for evaluation.

reality	forecast result	
	Predictive example	Predictive counterexample
Actual example	TP	FN
Actual counterexample	FP	TN

(10)

We use four indicators to evaluate the model， they are accuracy precision, recall, and f1score.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (11)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (12)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (13)$$

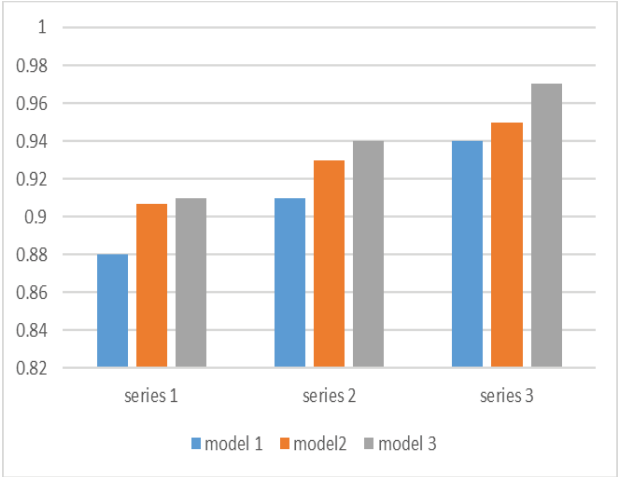
$$\text{F1score} = \frac{2*Precision*Recall}{Precision+Recall} \quad (14)$$

The evaluation results shown in (15):

Algorithm type	Precision	Recall	F1score
TF-IDF	0.9003	0.8981	0.8992
Merging Weighted TFIDF and word2vec	0.9233	0.9174	0.9203
Merging weighted improved TFIDF and word2vec	0.9426	0.9457	0.9441

(15)

The figure (16) reflects the accuracy of the three models in text classification of multiple categories.



(16)

The series 1, 2, and 3 respectively represent the text classification of the three models in two, seven, and fourteen categories. The model 1, 2, and 3 represent the traditional TFIDF model, the merged weighted TF-IDF and word2vec the merged weighted improved TF-IDF and word2vec model in turn. It is easy to conclude from the above data that the combined weighted model can significantly improve the classification accuracy of the traditional TF-IDF model. There is no significant difference in the classification accuracy between Model 2 and Model 3 when only two types of text classification are performed. In terms of multi-type text categorization, the improved TF-IDF evaluation method for keywords is more effective in multi-type text than the simple merged weighted model, which verifies the effectiveness of the improved model.

V. CONCLUSION

In view of the shortcomings of the traditional TF-IDF classification algorithm, this paper proposes a new improvement

method. By improving the TF, the probability that the unpopular vocabulary in the text is misclassified as a classification feature word is reduced. Considering the high frequency words in the sample class and the intra-class distribution bias leads to the problem that the multi-class text classification cannot be well performed, and the calculation method of IDF is optimized. At the same time, the semantic information of the keyword in the context is extracted by using word2vec, which further improves the accuracy of the classifier. How to train the word2vec model, make reasonable parameter adjustments to the skip-gram method and then further improve the classification accuracy is the next step to be studied.

REFERENCES

[1] J. Thorsten. Text categorization with Support Vector Machines: Learning with many relevant features[M]// Machine Learning: ECML-98. 1998.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] SALTON,G. A Vector space model for automatic indexing[J]. Communications of the Acm, 1974, 18(11):613-620.K. Elissa, “Title of paper if known,” unpublished.

[3] Ramanathan V, Meyyappan T. Survey of Text Mining[M]. 2004.

[4] Teh Y W, Newman D, Welling M. A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation.[C]// Conference on Advances in Neural Information Processing Systems. 2006.

[5] Sun Y, Lin L, Tang D, et al. Radical-Enhanced Chinese Character Embedding[J]. Lecture Notes in Computer Science, 2014, 8835:279-286

[6] Kenter T, Borisov A, De Rijke M. Siamese CBOW: Optimizing Word Embeddings for Sentence Representations[J]. 2016.

[7] Sheikh I, Illina I, Fohr D, et al. Document Level Semantic Context for Retrieving OOV Proper Names[C]// IEEE International Conference on Acoustics. 2016.