Research on Chinese Short Text Classification of Bidding Project Names with Fusion Feature Item Category Distribution

To cite this article: Yan Feng and Gang Qian 2019 J. Phys.: Conf. Ser. 1176 042075

View the article online for updates and enhancements.
Research on Chinese Short Text Classification of Bidding Project Names with Fusion Feature Item Category Distribution

Yan Feng¹*, Gang Qian²
¹School of information engineering, Nanjing University Of Finance & Economics, Nanjing, China
²School of information engineering, Nanjing University Of Finance & Economics, Nanjing, China

*Corresponding author e-mail: 995261840@qq.com

Abstract. Within the framework of word2vec, aiming at the feature of Chinese bidding project names, this paper proposes a TF-IDF-CDW weighted word2vec model, which combines the Category Distribution Weight (CDW) of feature items to generate short text vectors of project names. The short text vector is constructed in three ways, namely the mean word2vec model, the TF-IDF weighted word2vec model, and the TF-IDF-CDW weighted word2vec model. Finally, the three models are applied to the text classification of bidding project names. The experimental results are compared to verify the effectiveness of the new method.

1. Introduction
Along with the acceleration of China's marketization process, more and more enterprises use online bidding to conduct procurement of goods and tender for engineering projects. Bidding information is often scattered across multiple websites and is difficult to find. The bidding information website is professional in collecting bidding information. Due to the title of the tender is short and the information is centralized, the manual processing cost is high and the efficiency is low. In order to enable the bidder to obtain the required bidding information and respond as soon as possible according to the category. The classification of Chinese bidding project names has become an urgent problem to be solved in the bidding information website.

The Vector Space Model (VSM) [1] is the most widely used text representation model proposed by Salton et al. in 1975. The traditional VSM has several major problems: the vector of text is sparse and the dimension is high; the relationship between words and words is not considered and the VSM does not perform well in the semantic representation of text. With the development of deep learning, in order to overcome the weaknesses of the VSM, HINTON et al. proposed the distributed representation [2], which solved the problem of high dimensionality and sparsity of the traditional VSM. Text represented by a distributed representation can use the similarity between vectors to represent the semantic similarity of the text. Word2vec [3,4] uses distributed representation to represent a word vector, which is a tool for converting words into word vectors. In a large number of short text experiments, Word2vec has shown excellent processing power, and has been extensively used in Chinese word segmentation [5,6], POS Tagging [7], sentiment classification [8,9,10] and syntax dependency analysis [8,11]. Word2vec is becoming more and more popular because of the ability to seek deeper feature representations for short text data.

The commonly used short text vector representation models based on word2vec are: the mean word2vec model [12] and the TF-IDF weighted word2vec model. However, these two models have
certain defects when applied to the short text classification of bidding project names. The mean word2vec model considers that each word in the short text has the same weight, so it can’t distinguish the importance of the words in the text; the TF-IDF weighted word2vec model alleviates this problem to a certain extent, but the IDF term in TF-IDF [13] does not take into account the distribution of feature items among different categories, so that the TF-IDF algorithm may give high weight to the rare words (not important words) which are uniformly distributed in each category. So the TF-IDF weighted word2vec model perform poorly in the bidding project name classification. To tackle this problem, this paper advances a TF-IDF-CDW weighted word2vec model with the category distribution weight (CDW), CDW is mainly composed of two parameters, namely, concentration Degree (CD) and Distribution Degree (DD). Finally, the validity of the improved method is proved by the classification effect.

2. Related research

To study the short text classification of bidding project names, the first to be solved is the problem of text representation. Text representation is a major focus and difficulty of text categorization. In short, it means to express text as a data format that can be processed by computers. The most common text representation currently used is the Vector Space Model [1] (VSM), which makes the text representation a vector for computer processing. In VSM, the text is formalized as a point in the multi-dimensional space, and the processing of text is converted into vector operations in vector space, which greatly reduces the complexity of the problem. However, the VSM has some problems, and it does not perform well in some tasks of natural language processing (NLP). High dimensionality and sparsity are the biggest drawbacks of VSM. High dimensionality usually means that the dimension of the text vector represented by VSM is large and usually achievable $10^5$, which is easy to cause “dimensionality curse”. Sparseness, usually using a text vector represented by VSM, will result in a large number of "0" elements, while non-zero elements are extremely rare. Obviously, for several tasks of NLP, sparse vectors are unacceptable. VSM is based on the assumption that the keywords are linearly independent. It only considers the statistical properties of the words in the context. The words and words are independent of each other, without considering the semantic characteristics of the words themselves, so VSM have several limitations.

Due to the shortcomings of VSM, HINTON et al. put forward the word vector represented by Distributed Representation [2], which maps the words into a low-dimensional, dense real vector space (the space size is generally 100 or 200). The closer the words with similar meanings are, the closer they are in space. The higher dimensionality and sparsity of the traditional VSM are solved, and the text vector represented by distributed representation can use the similarity between vectors to represent the semantic similarity of the text.

With the in-deep study’s developing [8,9], word vector representation based on Neural network for self-feature extraction has more and more attention from industry and academia. Based on forerunner's study, word2vec [4] raised by Mikolov et al. [3] in 2013.It adopts the word vector represented by Distributed Representation, which uses the relationship between the feature word and its context to represent the feature word as a low-dimensional real number vector. The syntactic and semantic information of the word will be expressed well [14]. There are two types of Word2vec models, namely the CBOW model and the Skip-gram model. The CBOW model uses the $e$ ($e$ is an integer) words before and after the current word $t$ to predict the current word $t$; while the Skip-gram model is just the opposite, it uses the word $t$ to predict each $e$ words before and after the word $t$. This paper adopts CBOW model.

At present, there are few methods for representing short text vector by Word2vec. In the framework of word2vec, this paper proposes a TF-IDF-CDW weighted word2vec model which combines Category Distribution Weight (CDW) for generating short text vectors of names. Compared with the popular short text vector representation, namely the mean word2vec model and the TF-IDF weighted word2vec model, looking for a more suitable processing method for the name of the bidding project.
3. Algorithm design

3.1 Mean word2vec model.
After pretreatment, each word in the bidding project name is trained into a word vector by word2vec, and then each word vector of the text is added, divided by the amount of words, as a short text vector representing the text. \( S_i \) represents the \( i \)-th text, \( K_i \) stands for the number of words in the \( S_i \) text, and \( W_i(j) \) represents the word vector of the \( j \)-th word in the \( S_i \) text. \( V_i \) stands for the \( i \)-th text vector.

\[
V_i = \frac{1}{K_i} \sum_{1 \leq j \leq K_i} W_i(j) \tag{1}
\]

3.2 TF-IDF weighted word2vec model.
TF in TF-IDF refers to the frequency of words, that is, the number of times a word appears in the text. The more times a word appears in a text, the greater their effect in the text. IDF refers to the inverse document frequency, that is, if the more frequently a word appears in text sets, the lower the distinguishing ability of the word, the less the text characteristics can be reflected. \( S_i \), \( K_i \), \( W_i(j) \) and \( V_i \) are the same as above. \( word_i(j) \) represents the \( j \)-th word in the \( S_i \) text.

\[
V_i = \sum_{1 \leq j \leq K_i} W_i(j) \times \frac{tf(word_i(j), S_i) \times idf(word_i(j))}{\sqrt{\sum_{word_i(j) \in S_i} [tf(word_i(j), S_i) \times idf(word_i(j))]^2}} \tag{2}
\]

3.3 TF-IDF-CDW weighted word2vec model.
Aiming at the defects of the above two methods and the characteristics of the bidding project name, the TF-IDF-CDW weighted word2vec model was introduced to integrate the distribution of feature item categories. There are two important indicators for measuring the contribution of words to short text classifications: Distribution Degree (DD), Concentration Degree (CD).

Distribution Degree (DD) refers to the degree of dispersion of words within a certain category. If the word appears uniformly in a certain category, not in the individual text of the category, but scattered in many texts of the category, then the word is considered to contain more classification information, the ability to distinguish text of each category is stronger. It is more valuable for classification. The formula is as follows: \( N_{C_i,t} \) is the number of texts containing the word \( t \) in the \( C_i \) Category, and \( N_{C_i} \) is the total number of texts in the \( C_i \) Category.

\[
DD(C_i, t) = \frac{N_{C_i,t}}{N_{C_i}} \tag{3}
\]

At the same time, it is necessary to consider the Concentration Degree (CD) of words. The degree of concentration between classes refers to the distribution of words between categories in the whole text sets. If words are more concentrated in a certain category, and rarely appear in other categories, then the word is considered to be very helpful in distinguishing text categories. The formula is as follows: \( N_{C_i,t} \) is the same as above, and \( N_t \) represents the total amount of all texts which containing the word \( t \) in the whole text sets.

\[
DC(C_i, t) = \frac{N_{C_i,t}}{N_t} \tag{4}
\]

The TF-IDF-CDW algorithm based on category distribution weight is as follows:

\[
CDW(t) = \max DD(C_i, t) \times DC(C_i, t) \quad (1 \leq i \leq |C|) \tag{5}
\]

\[
TF - IDF - CDW(t_i) = TF(d, t_i) \times IDF(t_i) \times CDW(t_i) \tag{6}
\]

\( S_i \) is the \( i \)-th text, \( K_i \) represents the number of words in the \( S_i \) text, \( W_i(j) \) represents the word vector of the \( j \)-th word in the \( S_i \) text, and \( word_i(j) \) represents the \( j \)-th word in the \( S_i \) text. \( V_i \) represents the \( i \)-th text vector.
\[ V_i = \sum_{1 \leq j \leq K_i} W_i(j) \times \frac{tf(word_i(j), S_i) \times \text{idf}(word_i(j)) \times CDW(word_i(j))}{\sqrt{\sum_{word_i(j) \in S_i} [tf(word_i(j), S_i) \times \text{idf}(word_i(j)) \times CDW(word_i(j))]^2}} \] (7)

The TF-IDF-CDW algorithm not only retains the core idea of the TF-IDF algorithm, but also considers the class distribution of feature items, reducing the unreasonable weighting.

4. Analysis of results

The data sets used in this experiment is crawled from the bidding information websites of government and various institutions. The data is true and valid. A total of 403,328 pieces of data are captured, and excluding the winning bid name and the nullified bidding name of the same project. There are 204,757 bidding projects name remaining. In this paper, the data are trained as the training corpus of word2vec, because the larger the amount of data, the more accurate the semantic information contained in the word vector. Table 1 is a test of the trained word2vec model. Note: The English words in Table 1 are translated from Chinese text.

| word2vec model Similarity test | TOP1 | TOP2 | TOP3 |
|-------------------------------|------|------|------|
| Printer                      | Copier 0.882753 | Projector 0.850591 | Computer 0.842929 |
| Virescence                   | Lawn 0.881011 | Parkland 0.858025 | Gardens 0.833426 |

From these 200 thousand data, 16000 data were selected for manual classification and annotation. The 16000 data are divided into 8 categories, 2000 for each category. There are eight kinds, including health care, garden landscape, site engineering, home decoration, digital technology, lighting, electric power, and vehicle.

First of all, the data need be pre-treated, mainly for word segmentation, stop words removing. The segmentation tool used in this paper is Jieba word segmentation. In order to eliminate the influence of the classifier on the experimental results, the classifiers adopt in this paper are KNN and SVM [15,16]. Word2vec is implemented by genism [17] open source software. All experiments were performed using five-point cross-validation. The data sets was randomly divided into five parts, four of which were taken for training each time, one part was tested, and then the average of the five classification results was taken as the final result.

The test results were evaluated by correct rate (p), recall rate (r), and \( F_1 \) value indicators. The results are as follows. C1, C2, C3, C4, C5, C6, C7, and C8 represent eight categories of medical and health, garden landscape, site engineering, home decoration, digital technology, lighting, electric power, vehicle and ship. avg represent macros average.

| mean word2vec model (%) | KNN | SVM |
|-------------------------|-----|-----|
| p   | r   | \( F_1 \) | p   | r   | \( F_1 \) |
| C1  | 78.63 | 74.84 | 76.69 | 76.69 | 81.54 | 80.09 |
| C2  | 85.94 | 74.62 | 79.88 | 83.55 | 78.61 | 81.00 |
| C3  | 79.53 | 78.82 | 79.17 | 82.52 | 82.76 | 82.64 |
| C4  | 78.71 | 83.46 | 81.02 | 87.41 | 74.31 | 80.33 |
| C5  | 82.89 | 74.45 | 79.44 | 79.27 | 81.67 | 80.45 |
| C6  | 76.92 | 74.21 | 75.06 | 78.13 | 78.83 | 78.48 |
| C7  | 85.87 | 73.66 | 79.30 | 82.97 | 77.59 | 80.19 |
| C8  | 78.88 | 82.90 | 80.84 | 79.46 | 81.12 | 80.28 |
| avg | 80.80 | 77.12 | 78.80 | 81.50 | 79.55 | 80.43 |
Table 3. TF-IDF weighted word2vec model (%)

| category | KNN     | SVM     |
|----------|---------|---------|
| p        | r       | $F_1$   | p        | r       | $F_1$   |
| C1       | 79.23   | 82.39   | 80.78    | 89.37   | 79.58   | 84.19   |
| C2       | 83.47   | 73.78   | 78.33    | 81.25   | 85.92   | 83.52   |
| C3       | 82.52   | 77.85   | 80.12    | 80.08   | 83.47   | 81.74   |
| C4       | 75.53   | 81.82   | 78.55    | 84.98   | 86.86   | 85.91   |
| C5       | 85.49   | 78.73   | 81.97    | 83.76   | 86.57   | 85.14   |
| C6       | 78.66   | 77.42   | 78.04    | 88.65   | 78.28   | 83.14   |
| C7       | 82.88   | 78.19   | 80.47    | 87.42   | 82.64   | 84.96   |
| C8       | 81.76   | 79.08   | 80.40    | 79.51   | 86.17   | 82.71   |
| avg      | 81.19   | 78.66   | 79.83    | 84.38   | 83.69   | 83.91   |

Table 4. TF-IDF-CDW weighted word2vec model (%)

| category | KNN     | SVM     |
|----------|---------|---------|
| p        | r       | $F_1$   | p        | r       | $F_1$   |
| C1       | 83.67   | 83.53   | 83.60    | 90.17   | 83.06   | 86.47   |
| C2       | 90.72   | 75.43   | 82.37    | 88.89   | 85.57   | 87.20   |
| C3       | 86.65   | 80.29   | 83.35    | 82.95   | 88.46   | 85.62   |
| C4       | 79.91   | 84.66   | 82.22    | 85.58   | 87.86   | 86.71   |
| C5       | 84.08   | 83.82   | 83.95    | 84.55   | 84.60   | 84.57   |
| C6       | 83.11   | 80.35   | 81.71    | 89.38   | 82.43   | 85.76   |
| C7       | 88.95   | 79.52   | 83.97    | 86.07   | 85.11   | 85.59   |
| C8       | 86.24   | 84.37   | 85.29    | 86.79   | 81.72   | 84.18   |
| avg      | 85.42   | 81.50   | 83.31    | 86.80   | 84.85   | 85.76   |

According to Table 2 and Table 3, in the case of using the KNN classifier, the accuracy of the TF-IDF weighted word2vec model is $0.39\%$ larger than that of the mean word2vec model, the recall rate increased by $1.54\%$, and the $F_1$ value rose $1.03\%$; in the case of using the SVM classifier, the accuracy of the TF-IDF weighted word2vec model is $2.88\%$ greater than that of the mean word2vec model, the recall rate rose $4.14\%$, and the $F_1$ value increased by $3.48\%$.

According to Table 3 and Table 4, in the case of using the KNN classifier, the accuracy of the TF-IDF-CDW weighted word2vec model is $4.23\%$ larger than that of the TF-IDF weighted word2vec model, the recall rate increased by $2.84\%$, and the $F_1$ value rose $3.48\%$; in the case of the SVM classifier, the TF-IDF-CDW weighted word2vec model is $2.42\%$ greater than the TF-IDF weighted word2vec model, the recall rate rose $1.16\%$, and the $F_1$ value increased by $1.85\%$.

---

**Figure 1.** Comparison of classification effects of different models
5. Conclusion
Aiming at short text categorization of Chinese bidding project names, this paper analyzes the shortcomings of the mean word2vec model and the TF-IDF weighted word2vec model in the application of bidding project names categorization, and puts forward the TF-IDF-CDW weighted word2vec model, which combines the distribution of feature item categories. The experimental results show that the proposed method is effective.

References
[1] SALTON G, WONG A, YANG C S. A Vector Space Model for Automatic Indexing [M]. Communications of the Acm. 1974: 613-620.
[2] Hinton G E. Learning Distributed Representations of Concepts [C] //Proceeding of the 8th Annual Conference on Cognitive Science Society. Amberst, USA: [s.n.],1986:1-12.
[3] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed Representations of Words and Phrases and Their Compositionality[A] Advances in Neural Information Processing Systems[C],2013: 3111-3119.
[4] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient Estimation of Word Representations in Vector Space[J]. arXiv preprint arXiv, 2013.
[5] Sun Y, Lin L, YangN, et al. Radical-Enhanced Chinese Character Embedding[J]. Lecture Notes in Computer Science,2014,8835:279-286.
[6] Mansur M, Pei W, Chang B. Feature-based Neural Language Model and Chinese Word Segmentation [C]//IJCNLP.2013:1271-1277.
[7] Zheng X, Chen H, Xu T. Deep Learning for Chinese Word Segmentation and POS Tagging[C]//EMNLP.2013:647-657.
[8] Socher R, Bauer J, Manning C D, et al. Parsing with Compositional Vector Grammars[C]//Meeting of the Association for Computational Linguistics.2013:455-465.
[9] Socher R, Perelygin A, Wu J Y, et al. Recursive Deep Models for Semantic Compositional over a Sentiment Treebank[C]//Proceedings of the Conference on Empirical Methods in Natural Language Processing(EMNLP).2013:1631-1642.
[10] Tang D, Wei F, Yang N, et al. Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification[C]//ACL.2014:1555-1565.
[11] Zhang M,Zhang Y,Che W,et al. Chinese Parsing Exploiting Characters[C]//ACL.2013:125-134.
[12] Xing C, Wang D, Zhang X, et al. Document Classification with Distributions of Word Vectors[C]//2014 Annual Summit and Conference Asia-Pacific Signal and Information Processing Association(APSIPA).IEEE,2014:1-5.
[13] Joachims T. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization [M]. Springer US,1997:143-151.
[14] MIKOLOV T, YIH W, ZWEIG G. Linguistic Regularities in Continuous Space Word Representations[A] HLT-NAACL[C], 2013: 746-751.
[15] YANG Y. Expert Network: Effective and Efficient Learning from Human Decisions in Text Categorization and Retrieval[A] Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval[C], 1994 13-22.
[16] JOACHIMTS. Text Categorization With Support Vector Machines: Learning with Many Relevant Features[M]. City: Springer Berlin Heidelberg, 1998.
[17] Genism. http://radimrehurek.com/gensim