Journal of Computational Information Systems 10: 21 (2014) 9301–9308
Available at http://www.Jofcis.com

A New Study Based on Word2vec and Cluster for Document Categorization ⋆

Yanhong YUAN, Liming HE, Li PENG, Zhixing HUANG*

School of Computer and Information Science, Southwest University, Chongqing 400715, China

Abstract

As the rapid increase of internet resources, Nature Language Processing (NLP), including document categorization, has become a widely researched problem. Many models have been built to handle it, such as n-gram and hidden markov model. All of these can be used to represent documents via extracting the semantic features, but the dimension disaster is yet an unsolved problem. Word2vec is a new method, having low dimension, which is able to represent the semantic information of words. Inspired by this, this paper propose a new representation of document. At first, by using the word2vec toolkit, the all words are expressed as a series of vector space. Then, the cluster would be obtained by computing the distance between two words. After that, we are able to utilize it to represent documents and classify them by SVM. The results of experiments show that this representation can describe the document accurately.

Keywords: Word2vec; Natural Language Processing; SVM

1 Introduction

With the rapid development of information and technology, the popularity of the internet has caused an exponential increase in data and resources. In order to manage and utilize these effectively, data mining and information retrieval of content-based is becoming a popular filed, and document categorization, as a supervised learning task, is an important part of these. Giving the text categories, its main task is to identify the category what the text belongs to in the light of the context. With the massive growth of text data [1], text categorization is facing an unprecedented challenge. It is impossible to label these categories artificially. Hence, designing a fast and effective method for text classification has become a vital problem.

There is no essential differences between document categories and other classification problems. In a word, all problems of classification is to extract representative features of the training data and to match the corresponding label [2]. In general, researchers mainly consider the characteristics of two aspects in the text classification: semantic quality and statistical quality [3]. The most

⋆Project supported by the Natural Science Foundation Project of CQ CSTC (No. CSTC2012JJB40012), and the Fundamental Research Funds for the Central Universities (No. SWU1309265, XDJK2014B012).

*Corresponding author.
Email address: huangzx@swu.edu.cn (Zhixing HUANG).

1553–9105 / Copyright © 2014 Binary Information Press
DOI: 10.12733/jcis12500
November 1, 2014

common method of text classification is based on words. That is to say, the document would be represented as a vector which contains semantic and syntactic information of words. Vector space model (VSM) [4], which is a sample and can represent the semantic information accurately, is the most widely used document representation model. However, it still has some shortcomings. If the document is too long, the dimension of representation will be too large. In this case, the VSM will result in dimension disaster and cause serious impact to the text classification.

In order to avoid the curse of dimensionality, this paper proposes an alternative method of document representation. At first, every words are represented as a continuous space vector through adopting "word2vec", a deep learning toolkit. Then, these words will be clustered and new label added. After this, the document will be depicted by counting the number of the words. In this way, we can obtain the space vector of document and regard this representation as feature to train the classifier. We will use the SVM to get satisfactory results. A lot of experiments show that, this method will represent documents more effective than the previous method.

The rest of this paper is organized as follows. In Section 2, related works on text categorization and NLP are discussed. In Section 3, we introduce the details of the proposed method for text categorization. In the next section, we evaluate the proposed method through a lot of experiments. What's more, the conclusion and some ideas about future work are shown in Section 5.

2 Related Work

Natural Language Processing (NLP) has experienced a long history of research. In spite of this, NLP is still facing many problems due to the homonymy and polysemy. To date, there are research methods [5–7] to solve the problems of text classification. The most common method is representing document as a semantic digital vector and inputting this to a classifier for training and classifying. N-gram model is the most classical document representation model. In which, it is a sequence of terms with the length of N and mostly words are taken as terms. [8] has implemented the method of N-gram for text categorization which can represent the semantic information of documents well. Unfortunately, because the impact of dimension, N-gram can only result a good performance in the short text not long. Some other methods are also applied in text classification, such as probabilistic classification vector machines (PCVMs) [9] and latent dirichlet allocation (LDA) [10]. They are generative probabilistic models for collections of discrete data such as text corpora and are proved they can achieve satisfactory results through a large number of experiments. Nevertheless, these techniques are still not suitable for large-scale data because these techniques require high cost with respecting to computation or time complexity.

Word2vec is an open source tool from Google which is able to represent the word as a vector effectively. [11] shows that their novel model architecture can compute continuous vector representation of words from every large data sets. Since it can represent the context words well and calculate the similarity between words accurately, a great deal of researchers have focused on researching word2vec to represent the semantic meaning of words [12, 13]. [14] has utilized word2vec framework to represent the words which appear in documents. The result of experiment show that their method can eliminate the ambiguity of words and do well in clustering and classification. However, if the description of documents are directly from the representation of words, these description will be too large and broad to classify documents well.

Bekkerman [15] used the word-cluster representation which is computed using the information

bottleneck method to generate a compact and efficient representation do documents and point out that this method can yield performance in text categorization. Motivated by this idea, this paper uses word2vec to represent word as a vector. This method can reduce the dimension as you need and avoid the dimension disaster which caused by the One-Hot Representation in NLP. Using the cluster to represent a document not only can describe the main topic but also reduce the dimension of description. It can improve the performance of classification effectively. [16] indicated that the cluster of words and the category of documents had some stability in classification task. This proves that our method is feasible and effective.

3 System Framework

Firstly, we encode every words using word2vec. Then, we would use these codes to cluster these words and label the corresponding cluster. Getting the cluster, what we should do is to count the number of words in each cluster and use it as the variable of the corresponding dimension. After that, these vectors which can represent each document would be regarded as the input of SVM classifier to train and test.

3.1 Word2vec

Paper [11] refers that word2vec maps word to a series of continuous vector, high-dimensional digital representation, through building a neural network. Most important of all that the representation can be used to compute the similarity between words well. Dating back to 1986, Hinton [17] has proposed distributed representation. The main idea is to map each word into a k-dimensional digital vector and determine the semantic similarity between words by computing the distance. Word2vec extends this idea and represents each words as a digital vector. These generated vectors are mainly based on two models: Bag-of-Words and Skip-Gram.

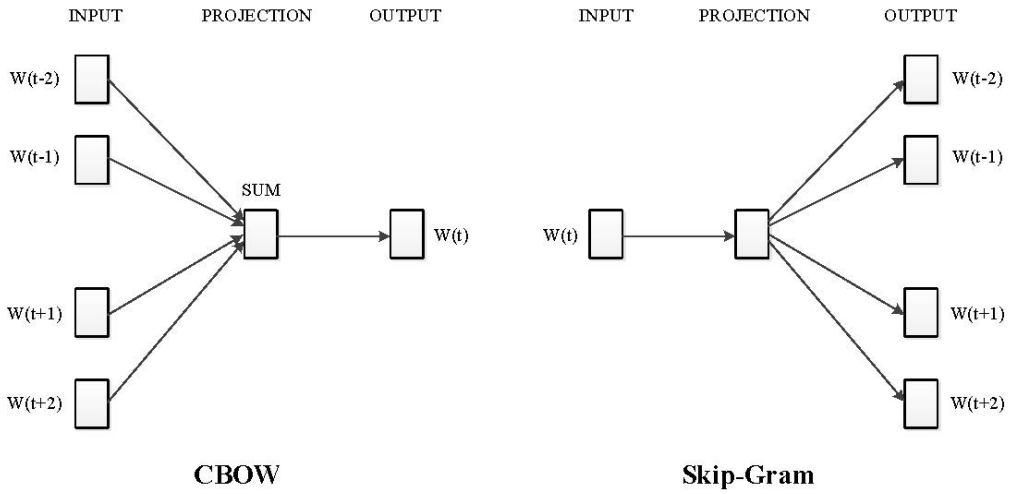

Fig. 1: The structure of Bag-of-Words and Skip-Gram

Bag-of-Words: This model is also called CBOW (Continuous Bag-of-Words model) because it uses the context of continuous distribution. The left of Fig. 1 shows the model structure. Seeing the structure, it is obvious that the hidden layer is a weighted sum of the context vectors which

belongs to the input layer. CBOW and the previous NNLM (Neural Network Language Model) has a same objective: predicting the probability of the next appearing word. The mathematical description is $p(w_t|w_{t-k}, w_{t-(k-1)}, \cdots, w_{t-1}, w_{t+1}, \cdots, w_{t+k})$. The difference is that CBOW remove the non-linear hidden layer which is the most time-consuming part and the projection layer which is shared for all words.

Skip-Gram: Skip-Gram is contrary to CBOW, predicting $p(w_i|w_t)$ with $t - c \leq i \leq t + c$ and $i \neq t$. The right of Fig. 1 shows the model structure. In 2006, Guthrie D et al. [18] introduced Skip-Gram model which reported for a certain skip distance k allow a total of k or less skips to construct the n-gram. The basic ideal of this model is generally summarized: we want to look at not only the set of adjacent words, but also sets of words where some are skipped. The most contribution is this model can skip some unimportant words to find the right phrase of words when these words are not continuous.

The generated vectors using the two models can express the semantic information of words. Especially in similar words, the most similar word can be found by computing the distance of vectors. Base on this ideal, we can calculate the cosine similarity between words to obtain similar words. Then, we would cluster these similar words and use the cluster to represent the semantic information of these words.

3.2 Cluster and document representation

In order to reduce the number of categories and improve the efficiency, clustering has become the first choice of researchers. The basic idea is to cluster the similar words into a class and use it to represent each words. As was mentioned above, the vectors computing by word2vec contain some semantic information. In word2vec, the biggest contribution is that we can accurately calculate the similarity between words using these vectors. For this, we can get the similar words by calculating their distance. In this paper, we chose the cosine similarity for calculating the distance. Two words are highly similar if their cosine similarity value is approaching 1. Therefor, we can cluster these similar words into a same class by computing their cosine similarity after we define the number of clusters.

After obtaining the cluster, we could count the appearing words in the cluster and use the number as the value of the corresponding dimension. Like this, the obtained vector is the feature of the document. This vector integrates all semantic information of similar words. Regarding these vector as the feature of documents and inputting into the svm classifier can find the similar documents accurately.

4 Experiments

In this section, we will introduce in detail how we design this method to achieve text classification and compare with other classical algorithms.

4.1 Dataset

In order to verify that our method is effective, we selected two standard data set of text categorization: UseNet news articles (20 Newsgroup [19]) and SRAA [20]. The 20 Newsgroup contains

over 20,000 newsgroup documents which are partitioned into 20 different newsgroup and we experiment with six of these categories. Table 1 shows six selected categories and the number of the text. SARR is a Simulated/Real/Aviation/Acto UseNet data which consists of 73,218 articles. It is often used for binary classification, where the task can be defined as the separation of documents on "real" vs. "simulated" or "auto" vs. "aviation". For balancing the number of documents in each category, we select only 5,000 documents in each category except "real-auto" which only contains 4,796 documents.

Table 1: The distribution of two data set

20 Newsgroup		SRAA	
CATEGORY	DOCUMENT	CATEGORY	DOCUMENT
Comp.graphics	973	Real-auto	4796
Rec.autos	990	Real-aviation	5000
Rec.sport.baseball	994	Sim-auto	5000
Sci.electronics	981	Sim-aviation	5000
Soc.religion.christian	997	–	–
Talk.politics.guns	910	–	–

4.2 The designs of experiments

It is necessary to preprocess the data set because the download text is not the standard format. First of all, we should remove some punctuation, digit and unimportant words.

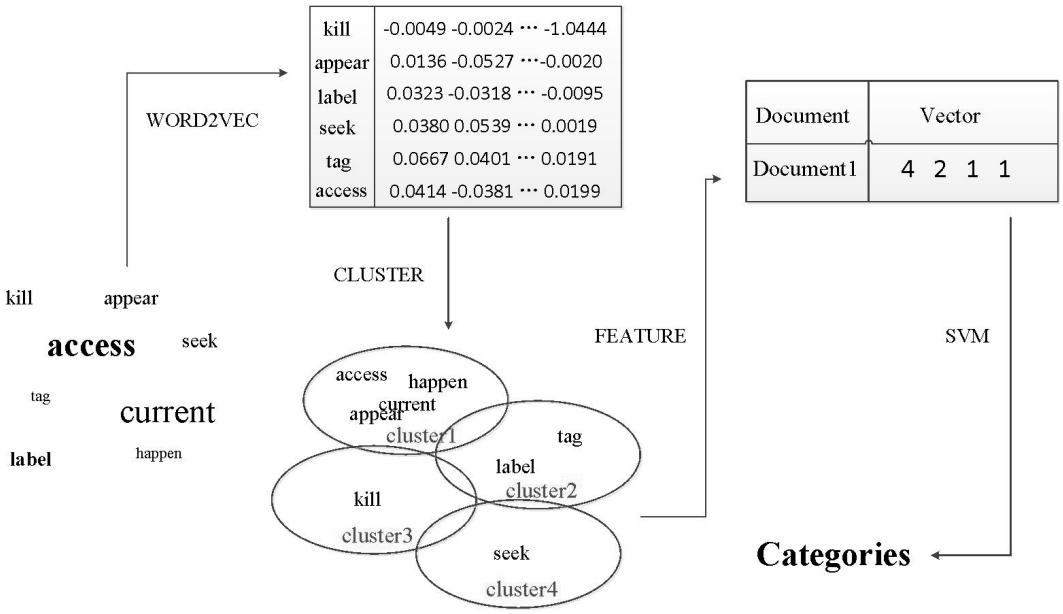

Fig. 2: The main process of experiments

After finishing the preprocessing, we regard these documents as a corpus. Each five continuous words is seen as a window and it will be pushed into word2vec which can computing the vector of

each word with the context. In this, the dimension of vector which we set is 200. We can obtain the $n \times 200$ vectors after training the neural network, in which n is the number of words. Getting these vectors, we would use these to compute the cosine similarity each two words. Then, the high similar words will be clustered to a cluster. In the design of experiments, we set the number of clusters is 50. In other words, we will get 50 different clusters and label them in each words. Next, we could compute the feature of document through these clusters: we initialize the features to a $d \times 50$ vector, where d denotes the number of documents and each column represents a cluster. If the word which belongs to i-th cluster appears in the document, the corresponding feature would plus 1 in the i-th dimension. With this method, we can obtain the all feature vectors.

We randomly select 70% of the data set to train the SVM classifier after obtaining the features, and the rest 30% to test. Final we can get the satisfied classification results. The main process is shown in Fig. 2

4.3 Results

To improve the accuracy of classification, we design and finish all experiments on Python 2.7 and MATLAB R2012b. The recognition ratio between the different categories in 20 newsgroup shows in Fig. 3 and the deep color represents a higher precision. We can see from the result that the precision in the data set is about 84%.

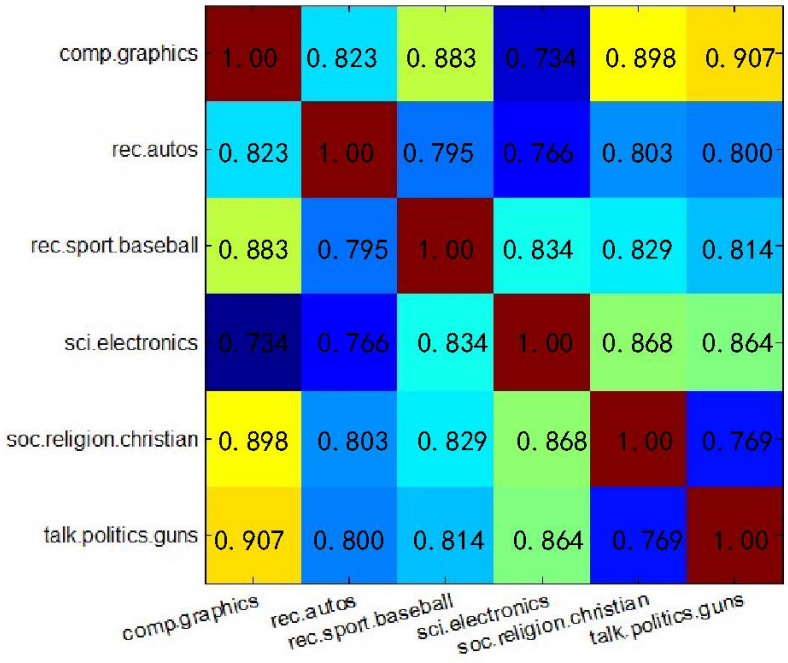

Fig. 3: The precision between the different categories in 20 newsgroup

In order to prove that it is effective for document classification through using of this method, we compare it with other traditional classification method, TF-IDF and LDA. Table 2 presented overall performance of several approaches conducted on each collection of documents, respectively. We measure the effectiveness of each method using the classical precision (P), recall (R) and F_1-measure, where F_1-measure is computed as $F_1 = (2PR)/(P + R)$. From the result, we can see

that our method can get more satisfactory results than other traditional method. From the Table 2 we can see that word2vec will show better performance for the second data set. The result also prove the theory that the word2vec can get better space vectors in big data set.

Table 2: The comparison between our method and IF-IDF

Datasets	Methods	Precision	Recall	F_1-measure
20 Newsgroup	TFIDF	0.597	0.640	0.616
	LDA	0.763	0.800	0.781
	OUR METHOD	**0.832**	**0.857**	**0.844**
SRAA	TFIDF	0.655	0.694	0.674
	LDA	0.827	0.843	0.835
	OUR METHOD	**0.931**	**0.945**	**0.938**

5 Conclusion and Future Work

In this paper, we obtain a number of clusters by computing the similarity between words and get the feature of document according to the number of the appearing words in each cluster. Then, the generated feature which are regarded as the input would be used for training a classifier. At last, we can find the right category for the unlabel documents. The results of these experiments show that this method can achieve a state-of-the-art classification.

However, this paper still has some shortage. For example, we only consider the vector of a discrete word rather than combine the context words for training the feature vectors. In the future work, we will focus on combining the context words to build a new code which may describe the semantic information of document better and improve the accuracy.

Acknowledgement

This work is supported in part by Natural Science Foundation Project of CQ CSTC (No. CSTC2012JJB40012), Fundamental Research Funds for the Central Universities (No. SWU1309265, XDJK2014B012). We also would like to thank the team members of Semantic Grid Research Group of Southwest University.

References

[1] Han E H S, Karypis G, Kumar V. Text categorization using weight adjusted k-nearest neighbor classification [M]. Springer Berlin Heidelberg, 2001.

[2] Liao Y, Vemuri V R. Use of K-nearest neighbor classifier for intrusion detection [J]. Computers & Security, 2002, 21 (5): 439-448.

[3] Hidalgo, Jos Mara Gmez. Text Representation for Automatic Text Categorization [J]. Eleventh Conference of the European Chapter of the Association for Computational Linguisitics, EACL. 2003.

[4] Dubin D. The most influential paper Gerard Salton never wrote [J]. 2004.

[5] Pilszy I. Text categorization and support vector machines [C]. The Proceedings of the 6th International Symposium of Hungarian Researchers on Computational Intelligence. 2005.

[6] Leopold E, Kindermann J. Text categorization with support vector machines. How to represent texts in input space? [J]. Machine Learning, 2002, 46 (1-3): 423-444.

[7] Lei S H I, Weng M, Xinming M A, et al. Rough set based decision tree ensemble algorithm for text classification [J]. Journal of Computational Information Systems6, 2010, 1: 89-95.

[8] Nther P. N-gram based Text Categorization [J]. Lomonosov Moscow State Univ, 2005.

[9] Chen H, Tino P, Yao X. Probabilistic classification vector machines [J]. Neural Networks, IEEE Transactions on, 2009, 20 (6): 901-914.

[10] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation [J]. The Journal of machine Learning research, 2003, 3: 993-1022.

[11] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space [J]. arXiv preprint arXiv: 1301. 3781, 2013.

[12] Mikolov T, Le Q V, Sutskever I. Exploiting Similarities among Languages for Machine Translation [J]. arXiv preprint arXiv: 1309. 4168, 2013.

[13] Zou W Y, Socher R, Cer D, et al. Bilingual Word Embeddings for Phrase-Based Machine Translation [C]. Conference on Empirical Methods in Natural Language Processing (EMNLP 2013). 2013.

[14] Wolf L, Hanani Y, Bar K, et al. Joint word2vec Networks for Bilingual Semantic Representations [J].

[15] Bekkerman R, El-Yaniv R, Tishby N, et al. Distributional word clusters vs. words for text categorization [J]. The Journal of Machine Learning Research, 2003, 3: 1183-1208.

[16] Zhuang F, Luo P, Xiong H, et al. Exploiting associations between word clusters and document classes for cross-domain text categorization [J]. Statistical Analysis and Data Mining, 2011, 4 (1): 100-114.

[17] Hinton G E. Learning distributed representations of concepts [C]. Proceedings of the eighth annual conference of the cognitive science society. 1986: 1-12.

[18] Guthrie D, Allison B, Liu W, et al. A closer look at skip-gram modelling [C]. Proceedings of the 5th international Conference on Language Resources and Evaluation (LREC-2006). 2006: 1-4.

[19] http://people.csail.mit.edu/jrennie/20Newsgroups/.

[20] http://www.cs.umass.edu/mccallum/data/sraa.tar.gz.