2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing

An Efficient Method for Document Categorization Based on Word2vec and Latent Semantic Analysis

Ronghui Ju*, Pan Zhou*, Cheng Hua Li† and Lijun Liu‡
*School of Electronic Information and Communications
Huazhong University of Science and Technology, Wuhan, China
Email: juronghui@hust.edu.cn, panzhou@hust.edu.cn
†DNN (Deep Neural Network) Lab, Beijing JingDong century trade .ltd (JD.COM). Email: lchbidatech@163.com
‡Wuhan TipDM Intelligent Technology, China. Email: liu_lijun_9@hotmail.com

***Abstract*—Document categorization is the process of classifying documents from many mixed documents automatically, and the main problem is how to express document content in vector space completely. This paper proposes a new model named Latent Semantic Analysis (LSA) + word2vec to categorize documents. This is the first attempt of combining word2vec with LSA at document categorization and it can map document to vector space under the premise of keeping document contents fully. At first, we create a term by document matrix and the element of which is decided by Term Frequency-Inverse Document Frequency (TF-IDF) weighting and word vector trained by word2vec. This matrix is a 3-dimensional matrix and it can describe the meaning of every word and the content of every document exactly. Secondly, Singular Value Decomposition (SVD) is executed on the matrix and lower computational complexity is gained from this. The model is named LSA + word2vec. Then, document vector gained from the new model are put into Convolutional Neural Network (CNN) to train. CNN is an efficient deep learning algorithm, which improves the accuracy of classification greatly. We evaluate the performance based on the 20newsgroups corpus. The results show that our new model achieves better effects on document categorization tasks, and the accuracy made about 15% improvement than traditional methods, such as LSA and Vector Space Model (VSM).**

***Keywords*—document categorization; word2vec; LSA; CNN**

I. Introduction

Nowadays, there is a great deal of information glutting our lives, especially on the Internet and most of which is text information. Document categorization is the first step to handle this big data problem for the purpose of large-scale information retrieval and fast text data processing. It can classify documents to one or more predefined categories according to their contents and it can understand the meaning of text, clarify contacts between documents. Consequently, document categorization has great significance in user behavior analysis, targeted advertising, text retrieval [1], email filtering [2], news classification [3] and so on.

In the past, research on text content extraction mainly used traditional methods such as the well known VSM [4] and LSA [5]. But they are all based on the frequencies of words or TF-IDF weighting. The obvious weakness of traditional methods is the elements of term by document matrix can not fully represent the meaning of words and the vectors consisting of them can not represent the documents perfectly.

Recently, the word2vec model and the CNN model bring up new opportunities in this field. Word2vec is an open source project proposed by Mikolov [6] who works in Google. It can map word to real vector and is considered as a perfect estimation of word representations in vector space. The main significance of this model is that it has large improvements in accuracy at much lower computational cost, so we can use vectors to represent words better. On this basis, there is a good method to represent documents by replacing the elements of term by document matrix as vectors created from word2vec. CNN is an efficient deep learning model and it has been demonstrated as an effective class of models for understanding text or image content, offering the state-of-the-art results on text detection, retrieval, content extraction. We use CNN to perform the final classification operation and get can much higher categorization accuracy.

In this work, we build a new LSA + word2vec model to convert documents to vectors and construct a specified CNN architecture to categorize them, which is the first try on combining LSA with word2vec and solves the problem of polysemy and synonymy existing in VSM. This model can not only raise about 15% accuracy than other traditional methods, but also reduce the computational complexity greatly. In details, we firstly use VSM to create the term by document matrix and we can use word coordinates to represent the document. The word vector read from Google News is laid up here to combine with the TF-IDF weighting to represent the weighting of word, so a document is expressed as 2-dimensional vector and it is a good feature vector representation to be categorized. Then, to make a better representation of documents, LSA is used in the process. LSA uses the vectors to represent the terms and documents as the same as VSM, but differently, it maps the terms and documents to latent semantic space thus it can remove some noise in the original vector space and improve the accuracy of information retrieval. Based on the term by document matrix created before, we execute SVD to reduce dimensions of vectors and get a more clear description of documents. Lastly, the new document coordinates are sent to a CNN model to train. The CNN architecture is designed specially for document classification tasks and is proved that it can make great performance on these tasks. So, a high accuracy of categorization is gained after the document vectors

978-1-5090-0154-5/15 $31.00 © 2015 IEEE
DOI 10.1109/CIT/IUCC/DASC/PICOM.2015.336

were trained.

The main contribution of this paper can be summarized as follows. In the first place, we propose a new method to create the term by document matrix based on word2vec, so that the meaning of document can be represented fully by vector. Secondly, we execute SVD on the new 3-dimensional matrix, so we can represent the latent semantic of documents perfectly. In the last, the 2-dimensional vectors of documents are put into the CNN to train. Our CNN architecture has two convolutional layers and two pooling layers, and document features are extracted through the training. The classification is operated based on the features and categorization accuracy is gained after this. We evaluate our new model based on the 20newsgroups corpus and achieve 50.4% accuracy, correspondingly, the accuracy of VSM is 33.6% and the accuracy of LSA is 34.2%. This is a big step forward on the document classification accuracy.

The rest paper is organized as follows. Section II discusses the related works of document categorization. Section III introduces our new model and its related background knowledge. Section IV describes convolutional neural network and shows our architecture. In section V, various results of this experiment are presented. Conclusion and future works are expressed in section VI.

II. Related Works

Document categorization consists in assigning predefined labels to text documents. It has great significance in processing network information and has prodigious economic value. For example, we can analyze user behavior based on web text browsing history and conduct targeted recommendation of contents or advertisements. In the past two decades, document categorization already had wide application in handling a large amount of text data. A considerable number of machine learning based approaches have been proposed. A great tutorial on the state-of-the-art of document categorization techniques can be found in [7].

In the document categorization task, many statistical classification methods and machine learning techniques have been used, including Rocchio classifier [8], K-Nearest Neighbor [9], Naive Bayes algorithms [10], decision trees [11], generative probabilistic classifiers [12], multivariate regression models [13] and VSM. VSM is a traditional and effective document categorization model and it becomes a general method to sort, retrieve and categorize documents since it was proposed by Salton et al. [4]. But it still has some problems, such as polysemy, synonymy and too high computational complexity. In order to reduce the feature vector representation and solve the problem of synonymy, many authors used the SVD technique in text categorization problems, such as [14] and [15]. LSA is the most representative method and based on this we have an effective way to express the meaning of document with vector. There are many works to categorize documents, such as [16], [17].

With the recent rejuvenation of neural networks starting with Hinton [18], deep neural networks have became a general and effective solution for extracting features, classifying data. The field of document categorization also has many tries on this. A hierarchical neural network system and a categorical neural network system for document classification was proposed by Chen et al. [19]. The automatically constructed thesaurus corresponding back-propagation neural network to vector space model based document categorization was proposed by Li et al. [20]. In addition, many document categorization tasks based on particular language was done, such as [21], [22] and so on.

In 2013, Google created a model that can make distributed representations of words and phrases and their compositionality named word2vec [23]. Based on this model, we can express a word or phrase's meaning with a vector, and we can calculate its relativity with others. Through the use of many people's works, word2vec is considered as an effective model for replacing a word's representation with vector. So, corresponding word2vec with VSM and LSA is reasonable and reliable.

Our work is inspired by [20] and [23] and combines advantages of LSA, word2vec and CNN. The new LSA + word2vec model solves the problem of polysemy and synonymy, gains much higher accuracy than traditional methods and reduces the computational cost greatly.

III. LSA + word2vec

A. Vector Space Model

The most important issue of document categorization is understanding the contents of documents. The simple method is using several words on behalf of the whole document to express the contents of it [24]. But only several words can not express the full meaning of documents. The basic idea of solving this problem is converting words or documents to vectors, and in vector space every word or document can be quantified and compared. In addition, the similarities between two words or documents can be calculated. For example, in the automatically constructed thesaurus [20], the top five most

TABLE I: The example of weighted terms associations

bank	Money	Rate	Dollar	Deposit	Rise
	0.5125	0.4045	0.4003	0.3315	0.3156
oil	Barrel	Crude	Energy	Petroleum	Gas
	0.5058	0.4338	0.3935	0.3762	0.3693
year	1986	January	Rise	Compare	December
	0.4726	0.3871	0.3771	0.3714	0.3705
price	Market	Higher	Low	Product	Steady
	0.4720	0.3537	0.3423	0.3322	0.2949
market	Price	Trade	Money	World	Coffee
	0.4720	0.3305	0.3183	0.2945	0.2897
billion	Rise	Money	Balance	Deposit	Cost
	0.4334	0.3881	0.3631	0.3482	0.3373
country	Economy	Develop	Debt	Foreign	Minister
	0.4074	0.4011	0.3129	0.3058	0.3027
council	Negotiate	Delegate	Minister	Republic	Session
	0.3684	0.3123	0.3114	0.2963	0.2809
forecast	Season	Harvest	Product	Output	Lower
	0.3613	0.3162	0.2608	0.2331	0.2229
agreement	Reach	Accord	Discuss	Sign	Meet
	0.3519	0.3427	0.2974	0.2883	0.2780

related terms in the Reuters-21578 corpus are associated with "bank", "oil", etc as shown in Table I.

Based on the similarity degree between words, we can calculate the difference of meaning. The similarity between documents is mainly affected by words that they contain. If we want to represent the documents based on vectors, the first thing to consider is the frequencies of words in documents. Therefore, VSM is proposed to represent the meaning of documents in vector space.

A vector space is a collection of objects called vectors, which may be added together and multiplied ("scaled") by numbers, called scalars in this context. Scalars usually are real numbers, sometimes are complex numbers or rational numbers and vectors are accepted, too. In vector space, a term by document matrix is made and the rows of which is words, the columns of which is documents. So, a document is represented as a vector, and each element of vector always is a value related to the words in this document. These vectors are columns of the matrix, TF-IDF weighting is a general value to compose the matrix as element of it.

TF-IDF is term frequency-inverse document frequency, and it can be expressed as follows:

$$\omega_{t,d} = tf_{t,d} \cdot log \frac{|D|}{|\{d' \in D | t \in d'\}|}, \tag{1}$$

where $tf_{t,d}$ is a term frequency of term t in document d, $log \frac{|D|}{|\{d' \in D | t \in d'\}|}$ is inverse document frequency. $|D|$ is the total number of documents in the document set, $|\{d' \in D | t \in d'\}|$ is the number of documents containing the term t. The word repeatedly appears in a document means that it is important for this document, but if it appears in many documents at the same time, this means that the word is a common word and it means a little for all of documents. Consequently, term frequency and inverse document frequency are used to calculate TF-TDF weighting at the same time.

Based on this, documents can be represented as follows:

$$d_j = (\omega_{1,j}, \omega_{2,j}, \ldots, \omega_{t,j}). \tag{2}$$

It is easier to calculate the cosine of angle between the document vectors:

$$cos\theta = \frac{d_1 \cdot d_2}{||d_1|| \cdot ||d_2||}, \tag{3}$$

and the similarity of documents can be represented by $cos\theta$. We can put the documents that have high similarity to one category together and document categorization is done.

B. Latent Semantic Analysis

In VSM, the frequencies of words appears in documents are considered as the feature of documents, and every document has exact words mapping. The meaning of documents is expressed as the weighting of words it contained. But because of the existence of polysemy and synonymy, VSM can not express the information of semantic level. For example, there are documents, one contains a word "automobile", and another contains a word "car", these two words will be considered as two different words, have different TF-IDF weighting and they maybe influence the meaning of documents. The worst thing is the two documents is classified as two categorizations because of the two words, after all, the two words has the same meaning. Consequently, LSA is essential.

The purpose of LSA is to find true meaning of words in documents and solve the problem above-mentioned. In details, it is using a model that has reasonable dimensions to express the words and documents for a large collection of documents. For instance, there is 1000 documents and 8000 words, LSA will create 100-dimensional space and mapping all of words and documents to this space. The procedure of mapping documents to this space is SVD and reducing dimensions. Reducing dimensions is the most important procedure. The noise is removed and the semantic architecture becomes clear by this operation.

In LSA, SVD of terms (words) by documents matrix can be formulated as follows:

$$C = USV^T, \tag{4}$$

where C is the term by documents matrix ($m \times n$). U is a $m \times m$ matrix and its columns is the orthogonal feature vectors of CC^T. V is a $n \times n$ matrix and its columns is the orthogonal feature vectors of C^TC. The feature values of CC^T and C^TC is same and it is $\lambda 1, \lambda 2, \ldots, \lambda n$. For S, S is a $n \times n$ matrix, $S_{ii} = \sqrt{\lambda_i}$, $\lambda_i > \lambda_{i+1}$ and zero otherwise.

To reduce the dimension of vector space to D, $S_{D+1,D+1}$ to S_{nn} are set to zero and S_{11} to S_{DD} are kept. After this, we can multiply U, S (having been reduced dimensions), V^T, and reconstruct the terms by documents matrix. In the new matrix, latent semantic of documents is presented, and we can calculate the similarity between documents or words much exactly. Each row of US is the term coordinate in latent semantic space and each row of VS is the document coordinate. The cosine angle between the coordinates of documents is considered as the classification standard in the neural networks.

C. word2vec

Word2vec has two novel model architectures for computing continuous vector representations of words from very large data sets [6]. The first architecture is Continuous Bag-of-Words Model (CBOW) that predicts the current word based on the words around it, and the second architecture is Continuous Skip-gram Model that predicts the surrounding words based on the current word (see Figure 1).

The training objective of CBOW is to find distributed word representation that can predict the current word based on words around it. It is similar to the feedforward neural network language model (NNLM) that was proposed in [25]. But it removes the most time-consuming non-linear hidden layer and only has 3 layers. The projection layer is shared for all words, and every word gets projected into the same position on the second layer. CBOW does not only use words from past, but also uses words from future. Unlike standard continuous bag of words model, it uses continuous distributed representation of the context [6]. So the best performance of predicting the current word based on i preceding words and i following words is obtained.

The purpose of Continuous Skip-gram Model is to find word representations that can predict the surrounding words

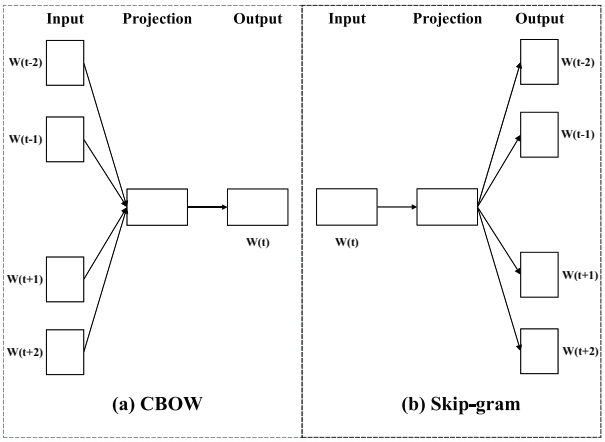

Fig. 1: Two model architectures of word2vec. The CBOW model is to predict the current word based on the words around it, and the Skip-gram model can find the most likely surrounding words based on the current word.

based on the current word. We define the training word as $\omega_1, \omega_2, \ldots, \omega_T$, the objective of the Skip-gram model is to maximize the average log probability

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(\omega_{t+j}|\omega_t), \tag{5}$$

where c is the size of the training context (which can be a function of the center word ω_t) [23]. Higher accuracy can be caused by larger c because larger c means more training examples, but it also can cause more expense of training time. The Skip-gram Model also removes the most time-consuming non-linear hidden layer and the current word is projected to the projection layer. The $p(\omega_{t+j}|\omega_t)$ can be calculated by the following softmax function:

$$p(\omega_O|\omega_I) = \frac{exp({\nu'_{\omega_O}}^T \nu_{\omega_I})}{\sum_{\omega=1}^{W} exp({\nu'_{\omega}}^T \nu_{\omega_I})}, \tag{6}$$

where ν_ω and ${\nu'}_\omega$ is the input and output vector representation of ω, and W is the number of words in the vocabulary [23]. Based on this, best performance of predicting the surrounding words is gained.

Through training, word2vec simplifies the context processing to vector processing in K-dimensional vector space. We can get the vector representations of words and the similarity between words can be calculated. The vector of words can be considered as a mapping from context space to vector space, and it can fully represents the words.

D. The new term and document relation matrix

The word vector trained by word2vec represents word meaning accurately, the TF-IDF weighting describes the influence of word on document. But only the set of word vector or TF-IDF weighting can not express the contents of document. Consequently, in this paper, TF-IDF weighting and word vector are used to represent document at the same time.

In our work, each element of term by document matrix in VSM is decided by TF-IDF weighting and word vector trained by word2vec. We multiply the word vector with the TF-IDF weighting and put the result to the term by document matrix to express the relation between the word and document. So, a $m \times n \times v$ (m is the number of words, n is the number of documents and v is the length of vectors trained by word2vec) matrix is created and a document is represented as a $m \times v$ matrix, as illustrated in Figure 2.

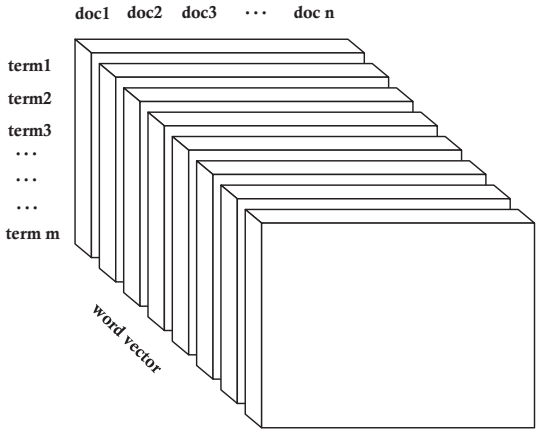

Fig. 2: The new term and document relation matrix

In the new 3-dimensional matrix, every document is represented as a matrix, each row is the product of TF-IDF weighting and word vector trained by word2vec. To represent the meaning of word completely, the word vector is trained by Google News, and it is considered as a standard word vector set. Every vector of word represents the meaning of it fully, TF-IDF weighting can express the importance of words in documents and 2-dimensional document vector express the document meaning exactly.

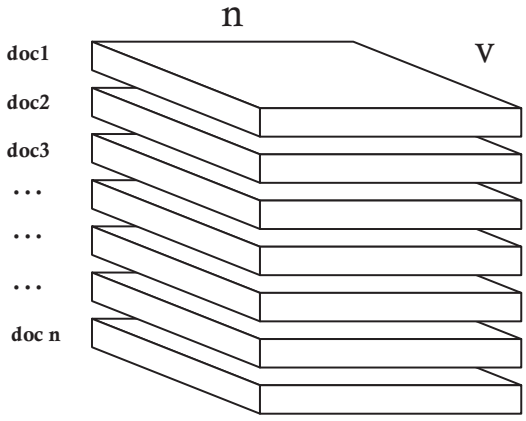

Fig. 3: LSA + word2vec model

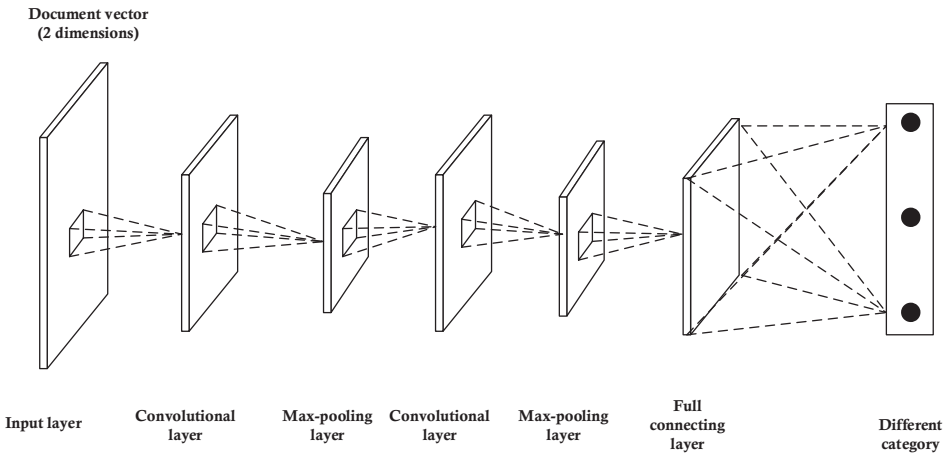

Fig. 4: The Convolutional Neural Network

E. LSA + word2vec

In this paper, after the new term and document relation matrix is created, SVD is done in each term by document 2- dimensional matrix. Every term by document matrix is decomposed as $C = USV^T$, and each row of VS matrix can distinguish documents as document coordinate. So, a 3-dimensional document coordinate matrix is constructed. In this matrix, each document coordinate is presented as a $n \times v$ matrix. This new document vector expresses the latent semantic of document and has every word vector that can represent word meaning fully in vector space. The structure of document coordinate is illustrated as Figure 3.

The new document coordinate matrix is a $n \times n \times v$ matrix. In the next procedure, the data set is put into a CNN model to classification and the categorization accuracy is calculated on the last step of it.

Based on the new document categorization model, the computational complexity of training on CNN is reduced a lot. In VSM, the term by document matrix is a $m \times n$ matrix ($m >> n$, m is the number of words, n is the number of documents). After combining with word2vec, the matrix becomes a $m \times n \times v$ matrix (v is the length of word vector) and it is a very huge matrix. SVD is executed and our new LSA + word2vec model is created. The new matrix is a $n \times n \times v$ matrix and it is much smaller than the $m \times n \times v$ matrix. So ,our new LSA + word2vec model not only improves the accuracy of categorization but also reduces the training time.

IV. Convolutional Neural Network

After we construct the new matrix of term and document relation and execute SVD to create the matrix of LSA + word2vec, all of document vectors are trained on a CNN architecture. The structure of CNN is shown in Figure 4.

This CNN structure is designed to categorize documents exactly, from the low-level local feature exaction layer to the high-level classification layer. The first layer is input layer, it contains matrixes (can be looked as 2-dimensional vectors) of documents, and the size of every matrix is $n \times n$ for LSA + word2vec model (n is the number of documents). After that, there are two convolution layers and each convolution layer is followed by a max-pooling layer to improve the robustness and reduce computational complexity.

In [26], the convolution operation is formulated as:

$$\mathrm{y}^{j(r)} = \max(0, b^{j(r)} + \sum_i k^{ij(r)} * x^{i(r)}), \tag{7}$$

where x^i and y^j are the i-th input map and the j-th output map. k^{ij} is the convolution kernel between the i-th input map and j-th output map. The symbol $*$ denotes convolution. b^j is the bias of the j-th output map. This formulation can fully express the convolution operation in CNN. Our max-pooling operation is similar to the expression in [26], but a little different. It is expressed as:

$$j^i_{j,k} = \max_{0 \le m,n < \min(a,b)} \{x^i_{ja+m,kb+n}\}, \tag{8}$$

where a is the height and b is the width of convolution kernel. y^i pools over an $a \times b$ non overlapping local region in the i-th input map x^i. In addition, we use ReLU for first convolution layer which has better fitting abilities, and we use sigmoid for second convolution layer.

After the second pooling layer, the connecting layer is needed. The classification operation is executed and the categorization accuracy is gained. The CNN architecture extracts the document features completely, and improves about 15% categorization accuracy which is significant progress in this field.

V. Experiments

A. Data sets for experiments

In order to evaluate the performance of new framework, 20newsgroups corpus is used on the experiments. Because of the limitation of the machine, we choose 3000 documents from 3 category and each category contains 1000 documents.

The 3 category is Comp.sys.mac.hardware, Rec.sport.baseball, Talk.politics.guns. The contents of these different categories are electronic products, sports, politics and they differences a lot and are easy to distinguish. We use 80 percent of documents to training and 20 percent for testing. This corpus is enough to evaluate the performance and obtain objective results.

The documents in the document corpus are general natural language text and they have many useless characters to understand the content of it. Consequently, the pretreatment of documents is necessary. We remove all of the punctuation, convert every capital letters to lowercase letters. After these operations, every document contains words only, and every letter is lowercase. The document corpus reduces the influence of the font format and the independent character. It is more easily handled and higher categorization accuracy is gained. The algorithms are written with Python.

B. Experiments of LSA + word2vec model

In our new LSA + word2vec model, every document in the document set is transformed to a 2-dimensional vector through our new LSA + word2vec model, and it can fully represent the document content. Then the document vectors are sent to the CNN architecture that we construct above. In the neural network, the number of input nodes is equal to the size of document coordinate ($n \times v$), and the number of output nodes is equal to the number of categorizes. After many iterations, the test accuracy is obtained. The overview of the architecture of our system is given in Figure 5.

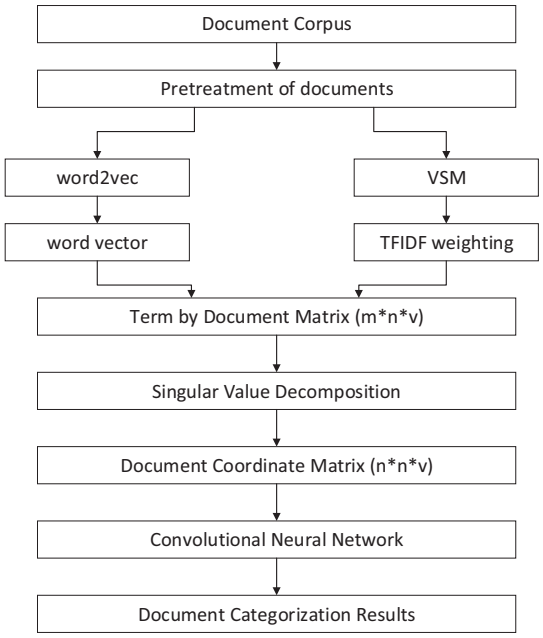

Fig. 5: System's architecture

The LSA + word2vec model is written in C++ and the CNN architecture is developed based on the caffe architecture. All of our experiments is conducted on Dell PowerEdge R720 sever that runs Intel Xeon processor and NVIDIA Tesla K20Xm GPU.

What we used to evaluate the new model's performance is the accuracy of categorization. In the output layer of our CNN model, a n-way softmax is used to predict the probability distribution over n different identities. In [26], the classification algorithm is formulated as:

$$\mathrm{y}_i = \frac{\exp(y_i')}{\sum_{j=1}^{n} \exp(y_j')}, \tag{9}$$

where $y'_j = \sum_{i=1}^{l} x_i \cdot \omega_{i,j} + b_j$, it is a linear function and combines the l features x_i as the input of node j. y_j is its output. After the classification function is created, test set is used to check the accuracy of categorization.

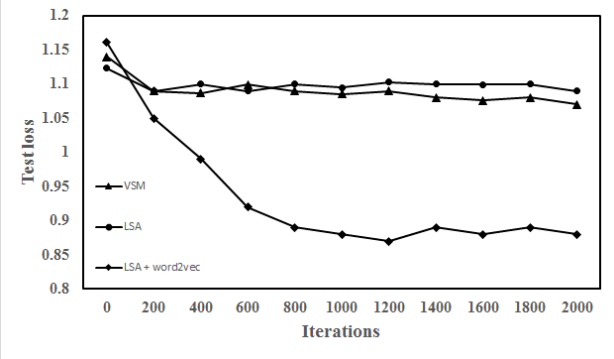

Fig. 6: Neural network test loss

The test loss is an important indicator in the performance evaluation of CNN. In our experiment, the document categorization test loss is showed in Figure 6. We can see that the test loss of VSM and LSA model is always high and hovering around 1.1, the test loss of our new LSA + word2vec model is down quickly at the beginning, and gradually stabilized at 0.87 at last. The lower test loss means higher accuracy, and from this Figure we can know that the LSA + word2vec model will get higher document categorization accuracy than VSM or LSA.

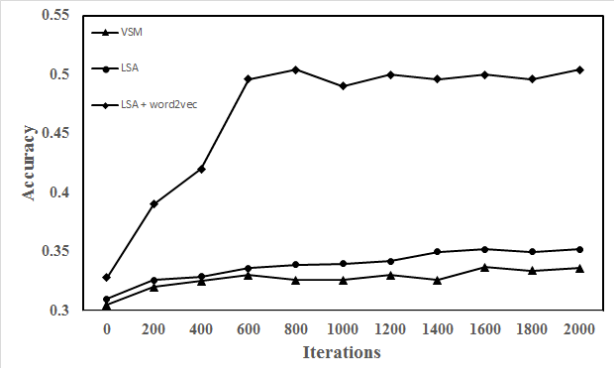

Fig. 7: Neural network test accuracy

In the case of test accuracy convergence, we can see details in Figure 7. At the beginning of phase, test accuracy of our new LSA + word2vec model is growing rapidly, and it slows down after a certain number of iterations, then shakes and levels off gradually. By contrast, test accuracies of the traditional document categorization model, VSM and LSA, grow slowly at the beginning, and then converge in a lower value quickly. The final result shows that our new LSA + word2vec model performs great in document categorization tasks and get a much higher categorization accuracy than traditional model.

TABLE II: Test accuracy of document categorization

Document categorization model	Accuracy (%)
VSM	33.6
LSA	34.2
LSA + word2vec	50.4

The Table II shows that the accuracy of VSM is 33.6%, the accuracy of LSA is 34.2%, and the accuracy of LSA + word2vec model is 50.4%. Our new LSA + word2vec model has better performance and get a about 15% higher document categorization accuracy than singular VSM or LSA. It means that our work makes significant progress in document categorization tasks.

C. The computation complexity of experiments

In the training process of CNN, the cost of training time and hardware resources is very huge. Consequently, reducing the training complexity is very important for our research and industrial applications.

To construct a document categorization model that can exactly express the meaning of words, combining VSM with word2vec is a simple and effective method. But there is a big problem that a document corpus has too many words and the document coordinate matrix is too big, so if we use these vectors for training in CNN it will spend a lot of time and occupy a lot of computing resources. Take the data set what we used in the experiment as an example, 20newsgroups corpus contains 18845 documents with a total of 0.7M words and a vocabulary size of 2K. The matrix is formulated as

$$Q = m \times n \times v, \tag{10}$$

where m is the number of the words, n is the number of the documents, v is the length of word vectors trained by word2vec and the dominating term is m. Then, the document coordinate matrix is trained in our specific CNN architecture, each input data is a $m \times v$ matrix. Because of the existing of convolution, back propagation and other parameter passing algorithm, a lot of time, computer memory and other computing resource are needed. Our new LSA + word2vec model provides a perfect solution to this problem. The operation of SVD is performed on the new term and document relation matrix and every document coordinate matrix is simplified to a small matrix. We use the new document coordinate matrix to train in the CNN. The matrix is trained on the CNN can be expressed as

$$Q = n \times n \times v (n << m), \tag{11}$$

where n is the number of the documents and it is much smaller than m, v is the length of word vectors trained by word2vec. Consequently, the computational complexity in the CNN model becomes very smaller than before. Our new LSA + word2vec model saves a lot of computing resources and computing time and it accelerates the progress of scientific research, reduces the costs and make industrial applying become much easier.

D. Analysis of experiments

In this section, we give the overall analysis of our proposed new LSA + word2vec model based on the experiment results above.

In the traditional models, VSM can only represent the effect of word frequencies in document content and meaning, and it can not express the phenomenon of polysemy and synonymy. This leads to the words that have the same meaning and different spelling is considered as totally different words, and the worst situation is it leads to the documents that contains these words are classified to different categories. The categorization accuracy of this model can reach 33.6% merely. Consequently, LSA is proposed to solve this problem and it execute SVD and reducing dimensions on term and document relation matrix. The generated document vectors and the word vectors can well represent the latent semantic of documents or words, and it makes computing the similarity between words and words, words and documents, documents become more reliable. To some extent, it has solved the problem of synonymy that Vector Space Model can not resolve and get 34.2% categorization accuracy. But there still are the problem of polysemy.

The word2vec model proposed by Mikolov [6] in Google provides an efficient estimation of word representations in vector space. After training this model on a big data set, every word can be represented as a vector and this vector can fully express the sense of word. Based on this model, the problem of polysemy has a good way to resolve.

Our new LSA + word2vec model combines advantages of VSM, LSA and word2vec and solves the problem of polysemy perfectly. Through combining VSM with word2vec, our new model expresses the influence of word frequency on document meaning and describes the word meaning accurately, so it solves the problem of polysemy. Then by means of SVD and reducing dimensions, it extracts the potential semantics of the documents and solves the problem of synonymy perfectly. Therefore, the document vectors generated by our new LSA + word2vec model can exactly express the content of documents and represent them in vector space completely.

The document vectors gained from our new LSA + word2vec model is sent to CNN to train and categorize. After two convolutional layers, two pooling layers and one full connecting layer, we get a good categorization model and this model get 50.4% categorization accuracy and it is about 15% higher than VSM or LSA.

VI. Conclusion and Future Works

In this paper, we propose a new model named LSA + word2vec model to create document vectors in vector space using the CNN model to get the accuracy of classification and evaluate the performance of categorization. Our new document vector matrix is created based on the subsets of 20newsgroups corpus. Our method can achieve 15% higher accuracy in document categorization than other general methods such as VSM and LSA. By combining with SVD and word2vec, each 2-dimensional document vector can fully represent the document in vector space space, and the computational complexity is reduced a lot. So, our new model makes great progress than other traditional model.

In the future, we will test our new LSA + word2vec model on Chinese document corpus and evaluate its performance. In Chinese corpus, what is needed to do firstly is participle, and maybe many details of the treatment are different from English corpus. It has great significance in the processing of non-alphabetic language text. Besides, we will try methods as much as possible to improve the accuracy of document categorization, not only in the stage of converting document to vector, but also in the classification stage using deep neural network.

Acknowledgment

This research is supported by NSFC No. 61401169.

References

[1] W. Lam, M. Ruiz, and P. Srinivasan, "Automatic Text Categorization and Its Application to Text Retrieval", IEEE Transactions on Knowledge and Data Engineering, vol. 11, pp. 865-879, 1999.

[2] J. D. M. Rennie, "An Application of Machine Learning to E-Mail Filtering", In Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Workshop on Text Mining, 2000.

[3] L. Wu, Z. Li, M. Li, W. Y. Ma, and N. Yu, "Mutually Beneficial Learning with Application to On-line News Classification", In Conference on Information and Knowledge Management Archive Proceedings of the ACM First Ph.D. Workshop in CIKM, pp. 85-92, 2007.

[4] G. Salton, A. Wong, and C.S. Yang, "A Vector Space Model for Automatic Indexing", Communications of the ACM, vol. 18, pp. 613-620, 1975.

[5] S. T. Dumais, "Latent semantic analysis", Annual review of information science and technology, vol. 38, pp. 188-230, 2004.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space", In Proceedings of Workshop at ICLR, arXiv preprint arXiv:1301.3781, Oct. 21, 2013.

[7] F. Sebastiani, "Machine Learning in Automated Text Categorization", ACM Computing Surveys, vol. 34, pp. 1-47, 2002.

[8] J. J. Rocchio, "Relevance feedback in information retrieval", In Gerard Salton (Ed.), The SMART Retrieval System: Experiments in Automatic Document Processing, Englewood Cliffs, News Jersey: Prentice-Hall, Inc. 1971.

[9] S. Tan, "An Effective Refinement Strategy for KNN Text Classifier", Expert Systems with Applications, Elsevier, vol. 30, pp. 290-298, 2006.

[10] D. D. Lewis, and M. Ringuette, "A Comparison of Two Learning Algorithms for Text Categorization", In Proceedings of 3rd Annual Symposium on Document Analysis and Information Retrieval, pp. 81-93, 1994.

[11] M. C. Wu, S. Y. Lin, and C. H. Lin, "An Effective Application of Decision Tree to Stock Trading", Expert Systems with Applications, Elsevier, vol. 31, pp. 270C274, 2006.

[12] D. D. Lewis, "The Independence Assumption in Information Retrieval", In Proceedings of the 10th European Conference on Machine Learning, pp. 4-15, 1998.

[13] J. Kazama, and J. Tsujii, "Maximum Entropy Models with Inequality Constraints: A Case Study on Text Categorization", Machine Learning, vol. 60, pp. 159-194, 2005.

[14] B. Yua, Z. Xub, and C. H. Li, "Latent Semantic Analysis for Text Categorization Using Neural Network", Knowledge-Based Systems, Elseiver, vol. 21, pp. 900C904, 2008.

[15] C. H. Li, and S. C. Park, "Combination of Modified BPNN Algorithms and an Efficient Feature Selection Method for Text Categorization", Information Processing & Management, Elseiver, vol. 45, pp. 329C340, 2009.

[16] P. Soucy, and G. W. Mineau, "Beyound TFIDF Weighting for Text Categorization in the Vector Space Model", Proceedings of the 19th International Joint Conference on Artificial Intelligence, pp. 1130-1135, 2005.

[17] M. Liu, and J. Yang, "An Improvement of TFIDF Weighting in Text Categorization", International Proceedings of Computer Science and Information Technology, pp. 44-47, 2012.

[18] G. E. Hinton, and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks", Science, vol. 313, pp. 504-507, 2006.

[19] Z. Chen, C. Ni, and Y. L. Murphey, "Neural Network Approaches for Text Document Categorization", International Joint Conference on Neural Networks, pp. 1054-1060, 2006.

[20] C. H. Li, W. Song, and S. C. Park, "An Automatically Constructed Thesaurus for Neural Network Based Document Categorization", Expert Systems with Applications 36, Elsevier, pp. 10969-10975, 2009.

[21] N. Chirawichitchai, P. Sa-nguansat, and P. Meesad, "Developing an Effective Thai Document Categorization Framework Base on Term Relevance Frequency Weighting", International Conference on ICT and Knowledge Engineering, pp. 19-23, 2010.

[22] M. EL Kourdi, A. Bensaid, and T. Rachidi, "Automatic Arabic document categorization Based on the Naive Bayes Algorithm", Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages, pp. 51-58, 2004.

[23] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality", Advances in NIPS, pp. 3111-3119, 2013.

[24] P. D. Turney, and P. Pantel, "From Frequency to Meaning: Vector Space Models of Semantics", Journal of Artificial Intelligence Research, vol. 37, Oct. 9, pp. 141-188, 2010.

[25] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model", Journal of Machine Learning Research, vol. 3, Jan. 3, pp. 1137-1155, 2003.

[26] Y. Cun, X. Wang, and X. Tang, "Deep Learning Face Representation from Predicting 10000 Classes", In Proc. CVPR, pp. 1891-1898, 2014.