Content Linking for UGC based on Word Embedding Model

Zhiqiao Gao, Lei Li

Beijing University of Posts and Telecommunications
Beijing, China
qiaogaozhi@bupt.edu.cn, leili@bupt.edu.cn

Liyuan Mao, Dezhu He, Chao Xue

Beijing University of Posts and Telecommunications
Beijing, China
753543585@qq.com, 723856523@qq.com,
memoryse7en@gmail.com

Abstract—**There are huge amounts of User Generated Contents (UGCs) consisting of authors' articles of different themes and readers' comments in social networks every day. Generally, an article often gives rise to thousands of readers' comments, which are related to specific points of the originally published article or previous comments. Hence it has suggested the urgent need for automated methods to implement the content linking task, which can also help other related applications, such as information retrieval, summarization and content management. So far content linking is still a relatively new issue. Because of the unsatisfactory of traditional ways based on feature extraction, we look forward to using deeper textual semantic analysis. The Word Embedding model based on deep learning has performed well in Natural Language Processing (NLP), especially in mining deep semantic information recently. Therefore, we study further on the Word Embedding model trained by different neural network models from which we can learn the structure, principles and training ways of the neural network based language models in more depth to complete deep semantic feature extraction. With the aid of the semantic features, we expect to put forward a new method for content linking between comments and their original articles in social networks, and finally verify the validity of the proposed method through experiments and comparison with traditional ways based on feature extraction.**

Keywords- Content Linking; Word Embedding model; UGC

I. INTRODUCTION

User Generated Contents (UGCs) have become the major component of social networks, and the scale of UGC has shown an explosive rise year by year.

Note that an author usually publishes an original article firstly in a social network, and then generally this article is followed or replied by a lot of readers, which are called comments or reviews. In these cases, we should pay more attention to the comments instead of neglecting them, because they can help people understand the original article more objectively.

TianYa (http://bbs.tianya.cn/) is a popular social network containing BBS, making friends and micro-blog, which is a very good UGC source for one to study. Focused on TianYa's corpora, this paper explores content linking between readers' comments and authors' articles or former readers' comments. It can also help other related applications, such as information retrieval, summarization and content management.

According to our former research about methods for content linking based on traditional feature extraction, we find that the problems mainly come from the disadvantage of traditional features that they simply deal with the surface features of grammars and semantics. Hence we plan to adopt the Word Embedding model based on deep learning so as to recognize the content linking with deeper semantic features.

The rest of this paper is organized as follows. Section 2 gives an overview of the related research. Section 3 presents the proposed method for content linking. Section 4 reports the experiments, results and analysis of Word Embedding model and content linking on real data collected from TianYa. Finally, Section 5 concludes this paper with future research issues.

II. RELATED WORK

Since UGCs' content linking is a very new research topic, there are little similar works. We will mainly discuss related works about Opinion Mining and Deep Learning.

In fact, the research for UGCs' comments have been carried out since they appeared, but it was mainly aimed for opinion mining of product information in the past, focusing on the study of emotional tendency[1].

Through the pretreatment of textual data and analyzing the content of product comments in various networks, we can find the consumers' attitudes and opinions to the commodities. And then, other consumers, manufacturers and retailers can obtain the feedback information they need, using the results of data mining and analysis of comments for commodities [2].

MultiLing 2015 (http://multiling.iit.demokritos.gr) started a task of content linking in social networks for the first time, putting the object for analysis to the direction of content correlation (which is also called content linking) and opinion mining: according to a news text and numerous comments, study the content correlation between comments and original contents or between comments and former comments, to exploit the specific twain statements of content correlation and identify the sentiment polarity of comment itself as well as the emotional consistency of this kind of opinion correlation.

Up to now, content correlation methods are mainly based on the multi-feature similarity calculation, using all kinds of existing classification and clustering methods based on rules or statistical models. But result of content linking we've got is not optimistic through traditional feature similarity calculation. So

we attempt to bring in deep learning technology applied to big data --- Word Embedding method to strengthen the traditional methods based on grammars and shallow semantics. The classic work is by Bengio et al. [3]. They used a three-layer neural network to construct the n-gram language model, and got word vectors through training the language model. From then on, many word vectors of different language models appeared constantly. In 2007, Mnih and Hindon[4] proposed a language model of Log-Bilinear. And then, they proposed a hierarchical idea to replace the most time-consuming multiplication of the matrix from the hidden layer to the output layer in Bengio's method. It ensured the effectiveness and enhanced the speed at the same time. Huang et al.[5] thought that it cannot exploit the semantic information of the target words fully only with the word vectors generated by local context information, so they attempted to give more semantic information to the word vectors. They proposed two major innovations: one is using the global information to assist the existing local information; the other is using multiple word vectors to represent polysemous words.

This representation learning has also been applied to a variety of natural language processing tasks with excellent results, such as Chinese word segmentation[6], semantic modeling and sentiment analysis[7], named entity recognition[8].

III. METHOD BASED ON WORD EMBEDDING MODEL

Figure 1 shows the framework of our proposed method for content linking based on Word Embedding model.

A. Pre-Processing

Due to the data for training and testing are crawled from the TianYa website, the first task is page cleaning and re-encoding.

We split paragraphs into sentences by some punctuations, such as " 。 ", " ！ ", " ？ ". We choose ICTCLAS (http://ictclas.nlpir.org/) for Chinese word segmentation.

B. Word Embedding Model

As to the calculation of text similarity, we usually regard the word as an important feature. The word vectors can map words from a single dimension to a vector space of K dimensions through training, and thus can seek deeper representation of semantic features for the textual data. The processing of the content is then simplified as the operation of vectors in a vector space of K dimensions, hence the similarity in the vector space can be used to represent the similarity of the textual semantics. Our experiments mainly focus on two types of training models: Word2vec and GloVe.
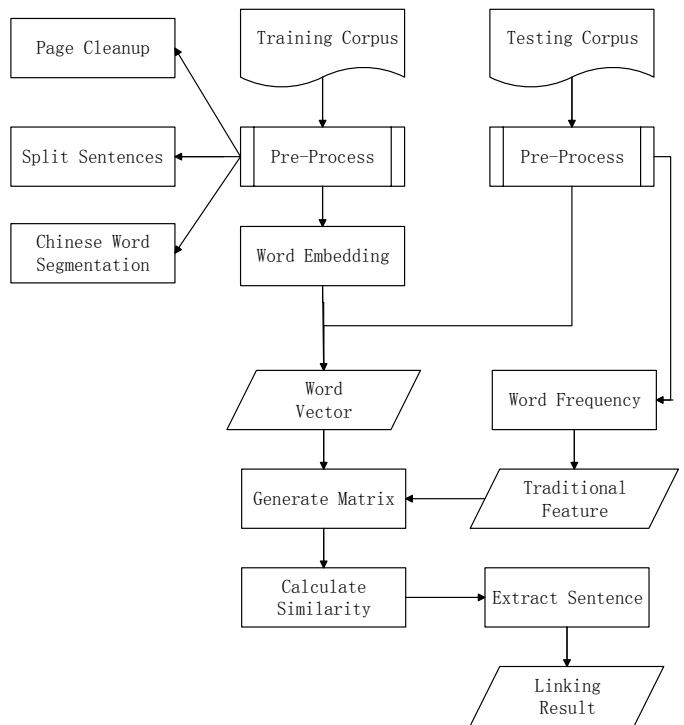
1) Word2vec

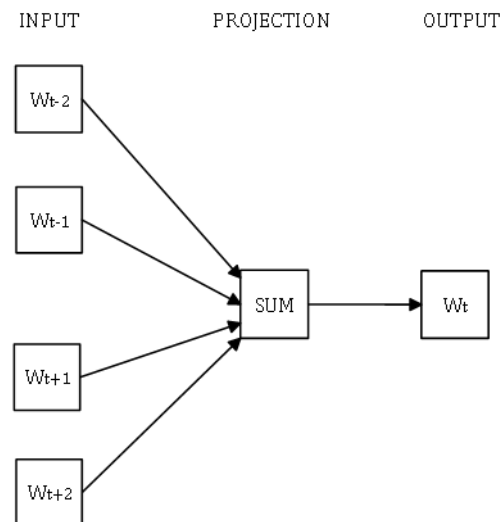Figure 1. Framework of the proposed method

Figure 2. CBOW

Word2vec is an efficient tool constructed by Google in the middle of 2013, achieving the model proposed by Mikolov et al.[8][9]. It uses the idea of deep learning to make the words be represented as real valued vectors by training. If we regard the word as the feature, then Word2vec can map the feature to a vector space of K dimensions.

Word2vec contains two types of training models: CBOW(Continuous Bag-of-Words Model) and Skip-gram. We use CBOW in our experiments. It is a kind of Hierarchical Log-Bilinear[10] language model as shown in Figure 2.

From Figure 2 we can see that CBOW model uses a neural network of three layers: INPUT-PROJECTION-OUTPUT.

INPUT selects a window of appropriate size as the context, and reads the corresponding word vectors; PROJECTION adds the word vectors (K dimensions with random initialization) read into the window together, forming a new vector of K dimensions, which is the K nodes in PROJECTION; OUTPUT is a huge binary tree, every none-leaf node here is a vector representing the word of a certain category, and every leaf node here represents a word vector. All the leaf nodes consist of all the words in the corpus, and the binary tree is constructed via Huffman Tree.

The mathematical equation for the model is:

$$P(w_t|\tau(w_{t-k},w_{t-k-1},\ldots,w_{t+k-1},w_{t+k})) \quad (1)$$

Where w_t represents a word in the dictionary. This equation uses the words in the window with size k adjacent to w_t to predict the probability of the word w_t . And function $\tau(w_1,w_2,\ldots,w_k)$ represents a certain operation with the parameters $w_i(1 \leq i \leq k)$. Word2vec uses the vector addition operation, which adds all the word vectors in the window together.

The objective function for CBOW is:

$$\frac{1}{T}\sum_{t=1}^{T}logP(w_t|\tau(w_{t-k},w_{t-k-1},\ldots,w_{t+k-1},w_{t+k})) \quad (2)$$

Where T represents the size of the dictionary. The target of the model is to maximize the value of this objective function. Adding *log* can transform the multiplication operation into the addition operation, which is convenient for subsequent operations.

2) GloVe

GloVe is a new global log-bilinear regression model for the unsupervised learning of word representations. It is proposed by Jeffrey Pennington, Richard Socher and Christopher D. Manning from Computer Science Department of Stanford University[11]. GloVe uses the statistics of word occurrences in a corpus which is the primary source of information available to all unsupervised methods for learning word representations. But it focuses on how meaning is generated from these statistics, and how the resulting word vectors might represent that meaning. For Global Vectors generated by GloVe, the global corpus statistics are captured directly.

The GloVe model is trained on the non-zero entries of a global word-word co-occurrence matrix, which tabulates how frequently words co-occur with one another in a given corpus. The point of GloVe is focused on the rate of word-word co-occurrence probability instead of the probability itself.

The calculation equation is as followed:

$$F\left(w_i,w_j,\widetilde{w}_k\right) = \frac{P_{ik}}{P_{jk}} \quad (3)$$

Let the matrix of word-word co-occurrence counts be denoted by X, whose entries X_{ij} tabulate the number of times that word j occurs in the context of word i. Let X_i be the number of times that any word appears in the context of word i. Let $P_{ij} = P(j|i) = X_{ij}/X_i$ be the probability that word j appear in the context of word i. Note that the ratio P_{ik}/P_{jk} depends on three words i, j, and k. And $w \in \mathbb{R}^d$ are word vectors and $\widetilde{w} \in \mathbb{R}^d$ are separate context word vectors.

Through a series of operation and simplification, we get the following equation:

$$w_i^T\widetilde{w}_k + b_i + \tilde{b}_k = log(X_{ik}) \quad (4)$$

Finally, adding an additional bias $\tilde{b}_k$ for $\widetilde{w}_k$ restores the symmetry.

It proposes a new weighted least squares regression model as following:

$$J = \sum_{i,j=1}^{V}f\left(X_{ij}\right)\left(w_i^T\widetilde{w}_j + b_i + \tilde{b}_j - logX_{ij}\right)^2 \quad (5)$$

Where V is the size of the vocabulary and $f\left(X_{ij}\right)$ is the weighting function.

For the word vector generated by GloVe, the Euclidean distance (or cosine similarity) between two word vectors provides an effective method for measuring the linguistic or semantic similarity of the corresponding words. And the result representations showcase interesting linear substructures of the word vector space.

C. Calculate similarity

After the training of word embedding models, a sentence in the testing corpus can be expressed as:

$$W_i = (w_t, w_{t+1}, \cdots, w_{t+k}) \quad (6)$$

Where w_t is word vector of corresponding word t. Then the sentence W_i and the sentence W_j can form calculating matrix $M_{i,j}$,

$$M_{i,j} = W_iW_j^T = \begin{bmatrix} w_tw_v & \cdots & w_tw_{v+l} \\ \vdots & \ddots & \vdots \\ w_{t+k}w_v & \cdots & w_{t+k}w_{v+l} \end{bmatrix} \quad (7)$$

The cosine distance can represent (w_t, w_v) , and the similarity of sentences i and j is as followed,

$$Sim_{i,j} = \frac{\sum_{m=i,n=j}max(M_{m,n})}{\sqrt{length_ilength_j}} \quad (8)$$

Where $max(M_{m,n})$ is obtained through the following concrete steps. First, find out the maximum of $M_{i,j}$, then delete the row and column of the maximum. Next, find the maximum of the remaining matrix and remove row and column like the former step. Do the same procedure until the matrix is empty. Finally add up all the maximum values. $length_i$ represents the number of word vectors in the sentence, and $\sqrt{length_ilength_j}$ is used to reduce the influence of sentence length.

D. Traditional features

As a comparison experiment, we use the traditional feature of word frequency to calculate the cosine distance between sentences via the traditional vectors in vector space model.

E. Linking results

Based on sentence similarities, we can extract those sentences with the highest similarity score with a comment sentence as its linking result. Note that there are many comments containing the same sentence that appears before in the article or former comments in our testing corpus. We just ignore them when evaluating our linking result, because they can be linked easily without difficult NLP analyzing.

IV. EXPERIMENTS AND RESULTS

A. TianYa BBS data

TianYa BBS, founded in March 1999, is a highly influential online community whose users' communication is mainly based on BBS forum, blog, etc.

There are huge amounts of UGC information and a lot of boards in TianYa. We have mainly worked on one of the most famous boards, namely TianYa Zatan Board, because it contains many hot topics discussed by plenty of users containing massive articles published originally and comments replied to them subsequently. Of course, there are also some comments replied to other former comments. One article and all its following comments can form a normal unit called a post.

In our experiments, the research corpus from TianYa website data is obtained automatically using a web crawler developed by ourselves. The corpus is in the standard format of BBS forum and the published dates are all in the period of more than three years, from 2012.1 to 2015.3. We have extracted complete information of articles and comments, including publishing time, publisher, number of comments, number of users replied, detailed contents of all comments, etc. The number of comments in the posts varies from 0 to 30,000. In fact, we find out that there may not be only one fixed theme in different comments even in just one post.

The size of training data is approximately 1G finally. When training the Word Embedding model, the corpus selected should be sampled evenly to avoid some missing in word vectors caused by the imbalance of the corpus.

According to the task of content linking, we have selected those posts whose comment number is greater than 30 in TianYa corpus as our testing data, since the post that was just followed by a few replies would not be much helpful for this experiment. Finally we have 3,502 posts for testing, including more than 50,000 sentences.

B. Word analogy task

To implement the comparison experiments for word vectors training on Word2Vec and GloVe, we set different sizes of Windows and dimensions. Due to the known expectation of the word vectors, vector(king) - vector(queen) + vector(man) = vector(woman), we can use this method to evaluate the result of trained word vectors.

We conduct experiments on the word analogy task of Mikolov et al.[9] The word analogy data set is just available in English. We firstly translated this data set into Chinese using Google translation, and then manually modified it. For example, there are some translated words that should be modified to commoner, and a number of verbs' tenses (such as run, ran, go,

went/gone, going, think, thinking), noun plurals (such as cat, cats, dog, dogs) as well as comparative adjectives (for example, short, shortest, slow, slowest), which do not exist in Chinese and thus should be deleted. The names of cities and provinces in China are also modified to their Chinese names, such as 沈阳 (Shenyang), 辽宁 (Liaoning), 哈尔滨 (Harbin), 黑龙江 (Heilongjiang).

TABLE I.   EXAMPLE OF THE CHINESE WORD ANALOGY DATASET

Common capital city	雅典(Athens) 希腊(Greece) 伦敦(London) 英国 (Britain)
Nationality	乌克兰(Ukraine) 乌克兰人(Ukrainian) 丹麦(Denmark) 丹麦人(Danish)
Currency	日本(Japan) 日元(yen) 美国(USA) 美元(dollar)
cities	沈阳(Shenyang),辽宁(Liaoning),哈尔滨(Harbin),黑龙江(Heilongjiang)
Man-Woman	叔叔(uncle) 姑妈(aunt) 儿子(son) 女儿(daughter)
Opposite	自在(free) 拘束(unfree) 已知(known) 未知(unknown)

Table I shows some examples of our Chinese Word Analogy Dataset that contains six types of semantic questions. Column 1 shows the semantic questions. Column 2 shows the corresponding word groups. Each line represents a word group containing various words separated by spaces (note that we have added the translations of Chinese words in its following brackets), which means that every two words are semantically related with the semantic relations listed in the same line in Column 1.

Finally, the Chinese word analogy dataset we obtained contains 11,214 word groups, and the training corpus covers 55.96% (6,275 word groups) of the dataset, as shown in Table II.

Question is assumed to be correctly answered only if the closest words to the vector computed using the above method (the known expectation of the word vector) is exactly the same as the correct word in the question. We evaluate the overall accuracy for each model with different parameters. In addition, the number of closest words N we set is not limited to 1, but expanded to 5.

TABLE II.   WORD ANALOGY TASK RESULT

Size-Window	GloVe	CBOW
50-15	22.96%	39.96%
100-10	36.36%	54.81%
200-8	46.56%	59.19%
Questions seen		55.96%

As we can see from Table II, CBOW performs better than GloVe in our experiments, and CBOW shows the best result when size=200, window=8.

Figure 3 gives some examples of our results from GloVe-100-10, in which each line represents a word group. As we can see, all these word groups can answer some of the semantic questions shown in Table I correctly.

多哈(Doha) 卡塔尔(Qatar) 马尼拉(Manila) 菲律宾(Philippines)

福州 (Fuzhou) 福建(Fujian) 济南(Jinan) 山东(Shandong)

......

哥哥(brother) 妹妹(sister) 新郎(groom) 新娘(bride)

男孩(boy) 女孩(girl) 爷爷(grandfather) 奶奶(grandmother)

希腊(Greece) 希腊人(Greek) 日本(Japan) 日本人(Japanese)

......

Figure 3.   Examples of our results from GloVe-100-10

Although we aimed to study content linking, we have also performed a simple comparison betweenWord2Vec and GloVe. Interestingly, GloVe was claimed to be better than Word2Vec in its papers, but our experiments have suggested its poorer performance in this Chinese word analogy task. Of course, there may be various influential factors for our experimental result, like different languages, corpus of different topics, and parameter settings, etc.

C. Content linking

Although we have enough data for testing, we have to evaluate the final result of content linking one by one through human labors. We have invited 5 graduate students to manually check the results. Finally, the evaluated testing data for content linking we used contains 30 posts consisting of 1,743 sentences in TianYa corpus. We have tagged the comment sentence linking with a score about the degree of its relativeness to the extracted sentence that the comment replied to. Scores range from 0 to 3, 0 indicates no association, the greater the score is, the greater the correlation. Finally, the average scores are shown in Table III.

TABLE III.   AVERAGE SCORES OF MODELS

Models	*Word2Vec*	*GloVe*	*Word Frequency*
Average Scores	1.059	0.998	1.029

As we can see from Table III, the final result suggests that Word2Vec performs the best, and followed by traditional feature method. It has also verified our assumption that using Word2Vec which produces deep semantic word vectors can provide a great deal of help to associated content task.

Table IV shows four examples of our linking results. For each example, line 3 gives the content of the comment sentence, line 1 gives the linking sentence for the comment sentence decided by Word Embedding model, line 2 gives the linking sentence for the comment sentence decided by tradition word frequency feature.

TABLE IV.   EXAMPLES OF LINKING RESULTS

	Example 1
Word Embedding	没有户口就没有责任田，没有宅基地 (If we have no registered permanent residence,we will have no responsibility fields or the homestead)
traditional feature	所以绝大多数超生户要给孩子办户口，中国的黑户绝不可能超过 1000 万 (So most families with more than one child have to handle registered permanent residence for their children, and the amount of the families without registered permanent residence in China can't beyond 10 millions)
comment	要像你这么说的黑户早饿死了哪还有黑户 (If it is like what you say, the families without registered permanent residence are starved earlier, where can we find them)
	Example 2
Word Embedding	我国的道路限速标识绝大部分与实际路况严重不符，完全可以跑 80 以上的路况、给你标上限速 40，完全可以跑 120 的高速路况、给你限速 60.....这样一来，在我国开车的任何驾驶员都不可能不超速 (the road speed limitation signs in our country are seriously discrepant with the actual road conditions, a road which can run more than 80 , it gives you a speed limitation to 40, a road which can run more than 120 , it gives you a speed limitation to 60...and then, it is impossible for any driver in China to run in the range of speed limitation)
traditional feature	交通管理部门可以堂而皇之的说为了安全，那不如限速 20 好了，我想应该就很安全了，当然也不绝对（限速 20 码的情况下也可能出交通事故的） (The traffic management department can say it is for security openly, while it is better to set the speed limitation to 20, I think it will be very safe, of course it's not absolutely(It may occur traffic accidents with the speed limitation to 20))
comment	我家门口一条路,江滨路，沥青路面，三车道，双向六车道，中间绿化隔离带至少 1.5 车道宽，两边还有一米左右的绿化带，绿化带外再是自行车道，车流不多（红绿灯都是一灯过），限速 40.超有才吧 (There is a road named Jiangbin Road in front of our house with asphalt pavement, three lanes, two-way six lanes, one middle green belt with at least 1.5 lanes width and one green belt with about 1 meter width in every side, and outside it is cycle path with very little traffic(The traffic light isn't busy).Isn't it a genius to set the speed limitation to 40?)
	Example 3
Word Embedding	中国有很多人有三个甚至四五个户口 (There are many Chinese with three or four or five registered permanent residence)
traditional feature	所以中国很多人都有双户口 (So there are many Chinese with two registered permanent residence)
comment	同意楼主意见，身边很多人都有两个以上的户口 (I agree with what the publisher said, there are many people with more than two egistered permanent residence around me)
	Example 4
Word Embedding	有了双户口，就可以多买房子，这一家人的名下一共有 11 套房子 (If we have two registered permanent residence, we can buy more houses,there are totally 11 houses in this family)
traditional feature	所以中国很多人都有双户口 (So there are many Chinese with two registered permanent

	residence)
comment	房妹家人是双户口的 (The families of the woman who have many houses have two registered permanent residence)

Let's just look at Example 1 for more detailed discussion. Although the comment and the linking sentence extracted by traditional features appear to have the similar key word "黑户 (family without residence registration)", we find out that its real meaning is the relation between residences with the fields through the context. Instead, the linking sentence by Word Embedding model shows the closer answer.

Realistically, we should point out some problems in this experiment that word vectors may sometimes lead to "excessive linking". It means that the word vectors can not only help to link two relevant sentences without the same vocabulary but also wrongly link two unrelated sentences by scoring them with much higher semantic similarity. We plan to work more on these problems via integration of Word Embedding model with traditional features and other possible features in future.

V.   CONCLUSION

In this paper, we mainly study the task of content linking between comment sentence and article sentence or former comment sentence in BBS post. Based on our former work of traditional features-based methods and its unsatisfied result, we propose to improve its performance by digging deeper semantic information with Word Embedding model. We then make further study on the Word Embedding model trained by different neural network models from which we can learn the structure, principles and training ways of the neural network language model in more depth to complete deep semantic feature extraction. With the aid of these semantic features, we implement a new method of content linking for interactive UGCs in social networks like BBS. Our experiments support a conclusion that the Word Embedding model based on deep learning performed well in deep semantics mining task as well as the content linking task by comparison with traditional ways based on feature extraction. There are still many issues for us to study more in future work, especially the possible improvement ways via integration of various methods so as to make full use of merits of existed technologies.

REFERENCES

[1] Pang, B., L. Lee. Opinion mining and sentiment analysis [J]. Foundations and trends in information retrieval, 2008, 2(1-2): 1-135.

[2] Li Shi, Ye Qiang, Li Yijun, R. Law. Mining features of product from Chinese customer online reviews [J]. Journal of Management Sciences in China, 2009, 315 12(2): 142-152

[3] Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." Science 313.5786 (2006): 504-507.

[4] Mnih, Andriy, and Geoffrey Hinton. "Three new graphical models for statistical language modelling." Proceedings of the 24th international conference on M Mnih, Andriy, and Geoffrey E. Hinton. "A scalable hierarchical distributed language model." Advances in neural information processing systems. 2009.achine learning. ACM, 2007.

[5] Huang, Eric H., et al. "Improving word representations via global context and multiple word prototypes." Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1. Association for Computational Linguistics, 2012.

[6] Lai Siwei, Xu Liheng, Chen Yubo, Liu Kang, Zhao Jun "Chinese Word Segment Based on Character Representation Learning." Journal of Chinese Information Processing 27.5 (2013): 8-14

[7] Socher, Richard, et al. "Semi-supervised recursive autoencoders for predicting sentiment distributions." Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2011.

[8] Turian J, Ratinov L, Bengio Y, et al. A preliminary evaluation of word representations for named-entity recognition[C]//NIPS Workshop on Grammar Induction, Representation of Language and Language Learning. 2009: 1-8.

[9] Tomas Mikolov,Kai Chen,Greg Corrado,et al. Efficient estimation of word representations in vector space [BE/OL].[2014-09-19].http://arxiv.org/abs/1301.3781v3

[10] Mikolov T,Sutskever I,Chen K,et al. Distributed representations of words and phrases and their compositionality[C]//NIPS,Lake Tahoe,USA,December,05-08,2013

[11] A. Mnih and G. Hinton. Three new graphical models for statistical language modelling. Proceedings of the 24th international conference on Machine learning, pages 641–648, 2007

[12] Pennington J, Socher R, Manning C D. Glove: Global vectors for word representation[J]. Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014), 2014, 12.