Journal of Computers Vol. 27, No. 4, 2016, pp. 55-67
doi:10.3966/199115592016122704005

Content Linking for Online Forums via Word Embedding Model

Lei Li[1*], Zhi-Qiao Gao[1], and Li-Yuan Mao[1]

[1] Department of Intelligence Science and Network Engineering, Beijing University of Posts and Telecommunications
Beijing100876, China
{leili, qiaogaozhi, circleyuan}@bupt.edu.cn

Received 20 April 2015; Revised 29 September 2015; Accepted 6 October 2016

Abstract. As a kind of important social network, online forums with rich and interactive user-generated contents (UGCs) have shown an explosive rise year by year. Generally, an originally published article often gives rise to thousands of readers' comments, which are related to specific points of the article or previous comments. This has formed the links of contents, which can provide a very good communication channel between publishers and their audience. 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. Up to now, most of the methods used for content linking are focused on similarity computing based on various traditional grammatical and semantic features. The major problem comes from the disadvantage that they mainly deal with the surface features of texts and words. In order to solve this problem, we propose to adopt deeper textual semantic analysis in this paper. Recently, the Word Embedding model based on deep learning has performed well in Natural Language Processing (NLP), especially in mining deep semantic information. Therefore, we 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 based language models in more depth to complete deep semantic feature extraction. We then put forward a new method for content linking between comments and their original articles for online forums, and verify the validity of the proposed method through experiments and comparison with traditional ways based on feature extraction using two realistic datasets.

Keywords: content linking, NLP, online forum, word embedding model

1 Introduction

The development of social networks has attracted more and more gaze of people in recent years with massive user-generated contents (UGCs) as the major component of information sharing. As a kind of important social network, online forums with rich and interactive UGCs have also shown an explosive rise year by year. Note that usually an author / publisher publishes an original article firstly in an online forum, then this article is followed or replied by a lot of readers in general, which are called comments or reviews. For instance, a given news article can often give rise to thousands of reader comments—some related to specific points / sentences within the article, others that are replies to previous comments [1]. Fig. 1 shows an example. This has formed the links of contents, which can provide a very good communication channel between publishers and their audience. In fact, this is also an important feature of Web 2.0 applications, which has replaced prevalent one-way reporting from publishers to their readers in previous Web 1.0 networks. 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.

According to Aker et al. [2], several user groups in media business now rely on online commenting to build and maintain their reputation and broaden their readers and customer base. Unfortunately, in the present set up of online forums, comments are difficult to organize, read and engage with, which affects

* Corresponding Author

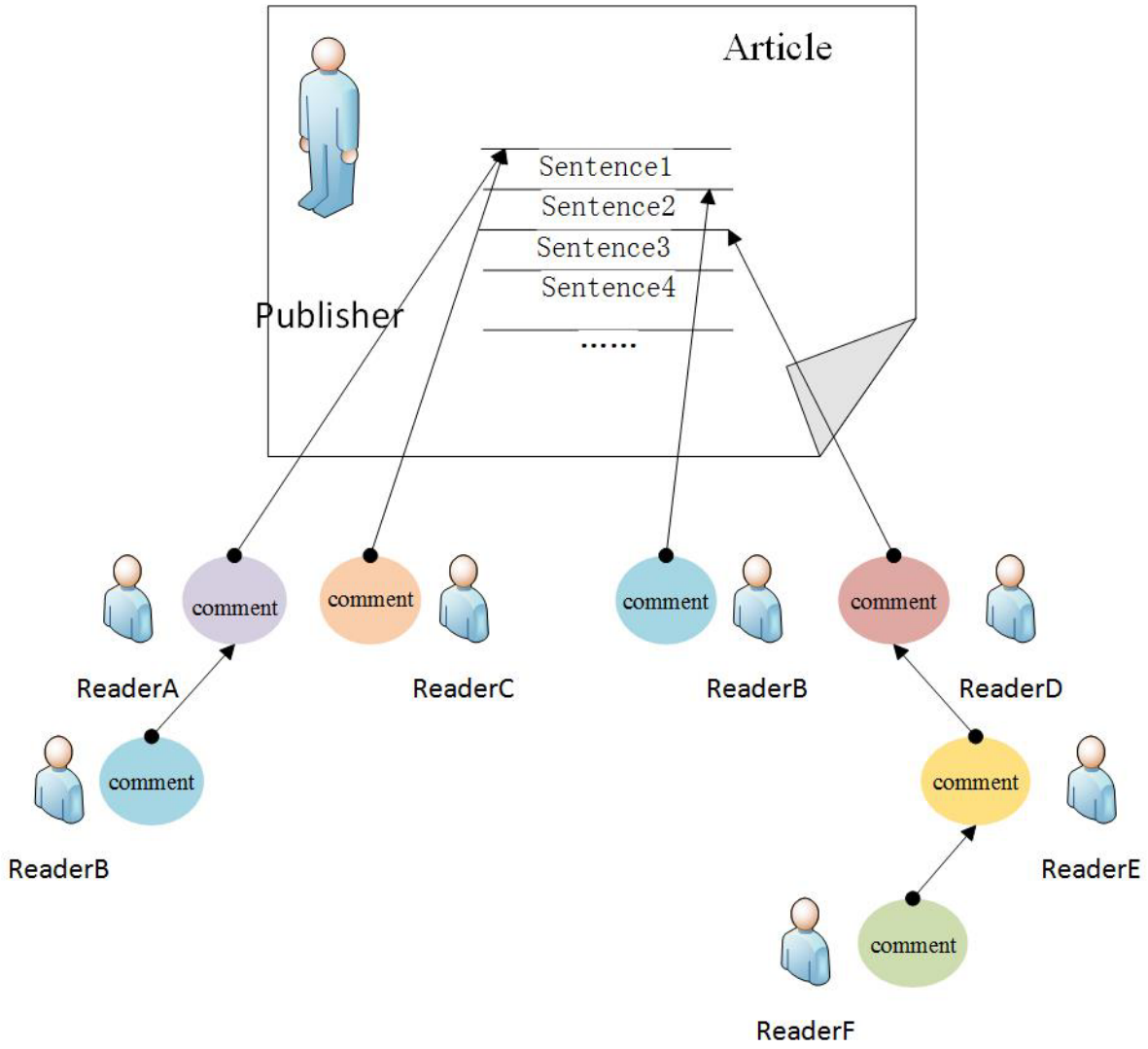

Fig. 1. Content linking in online forums

the quality of discussion and the usefulness of comments for the interested parties. One problem with comments in their current form is their detachment from the original article. Placed at the end of the article without clear reference to the parts of the article that triggered them, comments are hard to put into the context from which they originated, and this makes them difficult to interpret and evaluate. Hence, content linking built automatically between comments and articles is very useful for understanding online forums. It is also necessary in more complex systems such as information extraction, information retrieval, article / comment summarization and content management.

As far as we know, Online Forum Summarisation (OnForumS) pilot task [1] at MultiLing 2015 (http://multiling.iit.demokritos.gr/pages/view/1516/multiling-2015) is a pioneering attempt at setting content linking for online forums as one of its sub-tasks for automatic summarization and at bringing crowdsourcing to the evaluations. OnForumS 2015 has established itself as a valuable exercise in advancing the state-of-the-art in this new emerging area. According to its point of view, the high volume of such user-supplied comments suggests the need for automated methods to analyze these links, which in turn poses an exciting and novel challenge for the Natural Language Processing (NLP) community. The problem of producing a linking structure of such mass of comments touches on the area of text understanding in NLP, which leads to the task of content linking. Thus content linking task is to determine what comments link to, be that either specific points within the text of the article, or previous comments made by other users.

Up to now, most of the methods used for content linking are focused on similarity computing based on various traditional grammatical and semantic features, such as n-gram, word, term, named entity and co-

occurrence. We find out that the major problems come from the disadvantage that they mainly deal with the surface features of texts. Just as Tanev et al. [3] have also mentioned that lexical similarity cannot account for semantic similarities between different terms. For example, the phrases "expert in computer science" and "specialist in information technology" have no common terms, but in practice they are synonyms.

In this paper, we propose to adopt the Word Embedding model based on deep learning and combine it with some traditional features so as to recognize the content linking with deeper semantic features. We also design and implement a set of experiments to evaluate the performance of the proposed method using two realistic datasets. One is an English dataset provided by OnForumS 2015, which is collected from The Guardian, a very famous and major on-line news publisher in UK, who publishes articles on different topics and encourages reader engagement through the provision of an on-line comment facility. The other dataset is a Chinese one collected from TianYa (http://bbs.tianya.cn/) by us. TianYa is a popular online forum in China containing BBS (Bulletin Board System), making friends and micro-blog, which is a very good Chinese UGC source for one to study.

2 Related Work

Content linking for online forums is a very new research topic, we can find some closely related work from the OnForumS 2015 sharing task at MultiLing 2015. Kabadjov et al. introduces the OnForumS pilot task in details, including the task definition, data sets for training and testing, participating groups, and evaluation via crowd sourcing [1]. Aker et al. [2, 4] think that it is weak to use only lexical overlap between comments and source articles, so they investigate the effect of alternative representations of comments and news article texts on the results of comment-article linking with similarity metrics. They analyze the performance of the similarity method using terms, i.e., sequences of words which have all a meaning in a domain, and show that term based similarity linking outperforms similarity linking based on words. Krejzl et al. [5] think that the increasing amounts of user-supplied comments in most major online news portals bring a novel challenge for people. They use vector space model and latent dirichlet allocation (LDA) for comment linking. Tanev et al. [3] exploit an efficient algorithm for calculation of distributional similarity between pairs of terms, as well as term cooccurrences for content linking. In a word, researchers are trying to use various features to represent the contents and support similarity computing for recognizing content correlation.

Another related work about UGCs is mainly about "opinion mining". 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 [6]. 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 about their needs, using the results of data mining and analysis of comments for commodities [7]. OnForumS 2015[1] also has two sub-tasks of opinion mining. After the exploitation of the specific twain statements of content linking, a second sub-task is to identify the sentiment polarity of comment itself, and a third sub-task is to recognize the emotional consistency of this kind of opinion correlation. The major research strategy is to combine emotional knowledge and textual information for feature extraction, and to utilize various machine learning algorithms for model training and testing.

Up to now, we can see that both of the content linking methods and opinion mining methods are mainly based on multi-feature extraction of UGCs, since most researchers realize that only with good features can we make computers to grasp the meaning of the natural language text precisely. But the original result of content linking we've got is not optimistic, which mainly uses similarity computing based on rules or statistical models with traditional features. 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 for Word Embedding model is by Bengio et al. [8]. 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 [9] proposed a language model of Log-Bilinear. And then, they proposed a hierarchical idea [17] 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. [10] 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 [11], semantic modeling and sentiment analysis [12], named entity recognition [13].

3 The Proposed Method for Content Linking

Fig. 2 shows the framework of our proposed method for content linking integrating Word Embedding model with traditional features for both English and Chinese online forums.

3.1 Pre-processing

The processing unit is sentence, thus the first task for pre-processing module is to transfer the original format of the data set into a sequence of sentences. Then we need to obtain the information of words. Chinese language does not have natural spaces between words as English. We have implemented different pre-processing steps for English and Chinese forum data.

As to Chinese, due to the data for training and testing are crawled from the TianYa website using our own crawler, the first task is page cleaning and re-encoding. Then we split the original texts into sentences by some punctuations, such as "。", "！", "？". We choose ICTCLAS (http://ictclas.nlpir.org/) for Chinese word segmentation and remove stop words.

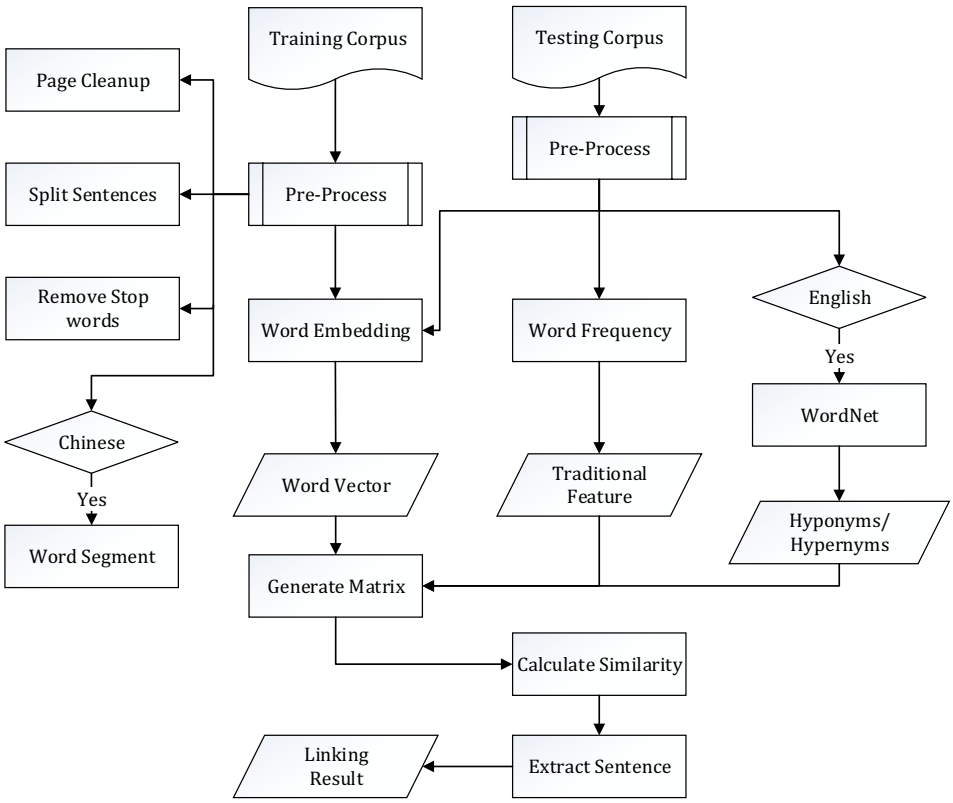

Fig. 2. Proposed framework for content linking in online forums

As to English, we use the testing data released by OnForumS 2015 collected from The Guardian. The original corpus of forums has two formats, i.e., txt and xml. Since xml format is easier to handle, we choose it and use DOM which is a toolkit in java to parse the xml file to get the all the sentences. An example of the xml data is shown as followed.

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE document SYSTEM "ofs.dtd">
<document id="37793736">
<articleText>
<s id="s0">Undervaluing Royal Mail shares cost taxpayers £750m in one day</s>
<s id="s1">The government 's desperation to sell Royal Mail cost taxpayers £750m in a single day, the National Audit Office has said in a scathing report into the privatisation of the 500-year-old national institution. </s>
<s id="s2">The public spending watchdog says the business secretary Vince Cable ploughed ahead with plans to float Royal Mail at a maximum price of 330p-a-share despite repeated warnings from City experts that the government had vastly undervalued the company. </s>
……
</articleText>
<commentaries>
<comment id="c0" bloggerId="richardbj ">
<s id="s41">[richardbj] And will anyone be sacked for incompetency and wasting public (yours and mine) money ? </s>
<s id="s42">Of course not.</s>
<s id="s43">Why not ?</s>
……
<comment id="c1" bloggerId="questionandfreedom ">
<s id="s49">[questionandfreedom] No one thinks that the tories will ever get back into government again, and Cameron knows that he has led the tory party into a political graveyard. </s>
</comment>
……
</commentaries>
<links>
</links>
</document>

Word Embedding model needs a large amount of textual data for training. Both of the sample data and testing data of OnForumS are too small in size. So we tried to collect the data ourselves for training corpus from Wikipedia (http://en.wikipedia.org/wiki/Wikipedia:Database_download) via a crawler. The size of our final training corpus is about 1G. Their pre-processing task is page cleaning and re-encoding, too. Then we split paragraphs into sentences by some punctuations, such as “.”, “!”, “?”, and split sentences into words by spaces. We also remove stop words.

3.2 Word embedding model

As to the representation of sentence and calculation of 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.

Word2Vec. Word2Vec is an efficient tool constructed by Google in the middle of 2013, achieving the model proposed by Mikolov et al. [13-14]. It uses the idea of deep learning to make the words be represented as real valued vectors by training.

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 [15] language model as shown in Fig. 3. 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)

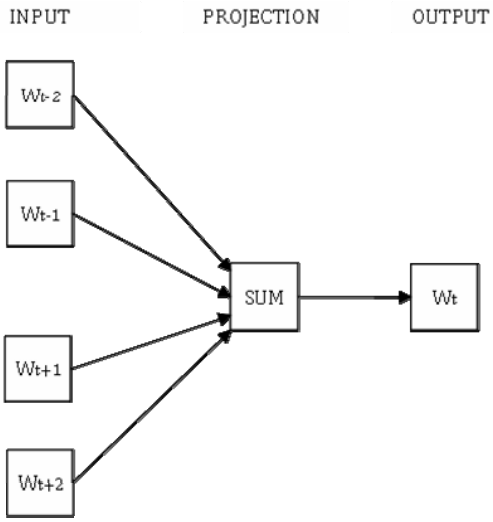

Fig. 3. CBOW

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:

$$P(w_t \mid \tau(w_{t-k}, w_{t-k-1}, ..., w_{t+k-1}, w_{t+k})) \tag{1}$$

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 $(w_1, w_2, ..., w_k)$ represents a certain operation with the parameters $w_i (1 \le i \le k)$. Word2Vec uses the vector addition operation, which adds all the word vectors in the window together.

The objective function for CBOW is:

$$\frac{1}{T}\sum_{t=1}^{T} \log P(w_t \mid \tau(w_{t-k}, w_{t-k-1}, ..., w_{t+k-1}, w_{t+k})) \tag{2}$$

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 one, which is convenient for subsequent operations.

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 [16]. 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:

$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}} \tag{3}$$

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 $P_{ij} = P(j \mid i) = \frac{X_{ij}}{X_i}$ be the probability that word *j* appears in the context of word *i*. Note that the ratio $\frac{P_{ik}}{P_{jk}}$ depends on three words *i*, *j*, and *k*. And $w \in R^d$ are word vectors and $\tilde{w} \in R^d$ are separate context word vectors.

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

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

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

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

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

where V is the size of the vocabulary and $f(X_{ij})$ 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.

3.3 Calculate similarity

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

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

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

$$M_{i,j} = W_i W_j^T = \begin{bmatrix} W_t W_v & \cdots & W_t W_{v+1} \\ \vdots & \ddots & \vdots \\ W_{t+k} W_v & \cdots & W_{t+k} W_{v+1} \end{bmatrix} \tag{7}$$

The cosine distance can represent $[\![(w]\!]_t, w_v)$, and the similarity of sentences *i* and *j* is as followed.

$$Sim_{i,j} = \frac{\sum_{m=i,n=j} \max(M_{m,n})}{\sqrt{length\boldsymbol{h}_i length\boldsymbol{h}_j}} \tag{8}$$

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. $length\boldsymbol{h}_i$ represents the number of word vectors in the sentence, and $\sqrt{length\boldsymbol{h}_i length\boldsymbol{h}_j}$ is used to reduce the influence of sentence length.

3.4 Traditional features

As a comparison experiment, we use the traditional feature of word frequency in Chinese to calculate the cosine distance between sentences via the traditional vectors in vector space model. For English, we also use a baseline system proposed in OnForumS 2015 [1].

3.5 Linking results

Aker et al. [2, 4] present the definition of content linking task as followed.

An original article in the online forum A is divided into n segments $S(A) = s_1, ..., s_n$. The article A is also associated with a set of comments $C(A) = c_1, ..., c_l$. The task is to link comments $c \in C(A)$ with article segments $s \in S(A)$. We express the similarity, or the strength of link between a comment c and an article segment s as their linking score (*Score*). A comment c and an article segment s are linked if and only if their *Score* exceeds a threshold, which we can experimentally optimize. When more than one s is linked with c, we can extract those sentences with the highest *Score* as the final linking result.

Note that there are many comments containing the same sentence that appears before in the article or former comments in our Chinese testing corpus. We just ignore them when evaluating the Chinese linking result, because they can be linked easily without difficult NLP analyzing.

4 Experiments and Results

We have implemented experiments on both the TianYa data in Chinese and the Guardian data in English released by OnForumS 2015.

4.1 The TianYa data and experiments

TianYa data. TianYa online forum, 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 TianYaZatan 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 January 2012 to March 2015. 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 the training data we used 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 imbalance.

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.

Word analogy task. To implement the comparison experiments for word vectors training on Word2Vec and GloVe, we set different sizes of windows and dimensions. According 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. [14]. 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 瀋陽(Shenyang), 遼寧(Liaoning), 哈爾濱(Harbin), 黑龍江(Heilongjiang).

Table 1 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 2.

Table 1. Example of the Chinese word analogy dataset

Type	Examples
Capital city-country	雅典 (Athens) 希臘 (Greece) 倫敦 (London) 英國 (Britain)
Nationality	烏克蘭 (Ukraine) 烏克蘭人(Ukrainian) 丹麥 (Denmark) 丹麥人 (Danish)
Currency	日本 (Japan) 日元 (yen) 美國 (USA) 美元 (dollar)
City-province	瀋陽 (Shenyang) 遼寧(Liaoning) 哈爾濱 (Harbin) 黑龍江 (Heilongjiang)
Man-Woman	叔叔 (uncle) 姑媽 (aunt) 兒子 (son) 女兒 (daughter)
Opposite	自在 (free) 拘束 (unfree) 已知 (known) 未知(unknown)

Table 2. Chinese 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**%

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.

As we can see from Table 2, CBOW performs better than GloVe in our experiments, and CBOW shows the best result when size=200, window=8.

Here are some examples of our results from GloVe-100-10, in which each line represents a word group.

多哈 (Doha) 卡塔爾 (Qatar) 馬尼拉 (Manila) 菲律賓 (Philippines)

福州 (Fuzhou) 福建 (Fujian) 濟南(Jinan) 山東 (Shandong)

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

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

希臘 (Greece) 希臘人 (Greek) 日本 (Japan) 日本人 (Japanese)

As we can see, all these word groups can answer some of the semantic questions shown in Table 1 correctly.

Although we aimed to study content linking, we have also performed a simple comparison between-Word2Vec 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.

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 3.

As we can see from Table 3, 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 3. Average scores of models

Models	Word2Vec	GloVe	Word frequency
Average Scores	1.059	0.998	1.029

Table 4 shows four examples of our linking results. For each example, column 2 gives the content of the comment sentence, column 3 gives the linking sentence decided by traditional word frequency feature, column 4 gives the linking sentence decided by Word Embedding model.

Table 4. Examples of linking results

#	Comment	Traditional feature	Word Embedding
1	要像你這麼說的黑戶早餓死了哪還有黑戶 (If it is like what you say, the families without registered permanent residence are starved earlier, where can we find them)	所以絕大多數超生戶要給孩子辦戶口，中國的黑戶絕不可能超過 1,000 萬 (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)	沒有戶口就沒有責任田，沒有宅基地 (If we have no registered permanent residence,we will have no responsibility fields or the homestead)
2	我家門口一條路，江濱路，瀝青路面，三車道，雙向六車道，中間綠化隔離帶至少 1.5 車道寬，兩邊還有一米左右的綠化帶，綠化帶外再是自行車道，車流不多（紅綠燈都是一燈過），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 setthe speed limitation to 40?)	交通管理部門可以堂而皇之的說爲了安全，那不如限速 20 好了，我想應該就很安全了，當然也不絕對（限速 20 碼的情況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))	我國的道路限速標識絕大部分與實際路況嚴重不符，完全可以跑 80 以上的路況、給你標上限速 40，完全可以跑 120 的高速路況、給你限速 60……這樣一來，在我國開車的任何駕駛員都不可能不超速 (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 aspeed 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)
3	同意樓主意見，身邊很多人都有兩個以上的戶口 (I agree with what the publisher said, there are many people with more than two egistered permanent residence around me)	所以中國很多人都有雙戶口 (So there are many Chinese with two registered permanent residence)	中國有很多人有三個甚至四五個戶口 (There are many Chinese with three or four or five registered permanent residence)
4	房妹家人是雙戶口的 (The families of the woman who have many houses have two registered permanent residence)	所以中國很多人都有雙戶口 (So there are many Chinese with two registered permanent residence)	有了雙戶口，就可以多買房子，這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)

Let's just look at Example 1 in Table 4 for more detailed discussion. Although the comment and the linking sentence extracted by traditional features appear to have the similar key word 「黑戶」(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 sen-

tences 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.

4.2 The Guardian data and experiments

The Guardian data. According to OnForums 2015[1], the official test data set in English consists of ten articles from The Guardian together with corresponding top fifty comments for each article (articles may contain thousands of comments), containing 43,104 words.

The approach used for evaluation in OnForumS 2015 is IR-inspired and based on the concept of pooling used in TREC[18], where the assumption is that possible links that were not pro-posed by any system are deemed irrelevant. Evaluation is based on the results of a crowdsourcing exercise. Contributors are asked to judge whether potential links are correct for each test article and its comments. This is a validation task as opposed to annotation, that is, contributors are only asked to validate links and labels produced by systems and are not asked to link or label data themselves. Hence there is only precision value. Additionally, due to the high volume of system links only a subset of all the links produced by systems is evaluated by extracting a stratified sample.

Word Embedding Model with WordNet. We use GloVe and Word2Vec for English Word Embedding Model, combined with WordNet to compute sentence similarity. WordNet is a semantics-oriented dictionary of English, similar to a traditional thesaurus but with a richer structure, which makes it easy to navigate between concepts. For example, given a concept like "car," we can look at the concepts that are more specific—the hyponyms: "Stanley steamer," "hardtop," "loaner" and so on. We can also navigate up the hierarchy by visiting hypernyms, like "car": "motor vehicle."

Word analogy task. For this dataset, we have obtained 19,544 word groups, and the English training corpus covers 99.70% of it, as shown in Table 5. This time the number of closest words *N* we set is 1. As we can also see, when size=200, window=8, models perform best in our experiments. Similarly, Word2Vec performs better than GloVe.

Table 5. English word analogy task result

Size-Window	GloVe	CBOW
50-15	26.95%	41.76%
100-10	38.92%	56.43%
200-8	**47.33%**	**60.29%**
Questions seen		**99.70**%

Content linking. Here before the computation of $[\![(w]\!]_t, w_v)$ in equation (7), firstly, the stemmed English words are generated and examined in consistency. Secondly, it is essential to check relations between word *t* and word *v* by WordNet. When word *t* and word *v* exist in the hyponyms / hypernyms part of each other, they can be seen as the same. Table 6 shows our experimental results.

According to the corpus of OnForumS 2015 and the evaluation method, we can implement experiments through a variety of thresholds for *Score* to check the performance of Word2Vec and GloVe, not like section 4.1 which requires human labor. As a comparison, we use the baseline system called Overlap [1], which links a comment sentence to the article (or parent comment) sentence with the most common words and gets the highest precision in OnForumS 2015.

As we can see from Table 6, the column "Threshold" shows three different thresholds for every content linking method, the column "Correction" indicates the number of correct linking, the column "Sum" indicates the total number of content linking found, and the column "Precision" gives the corresponding accuracy rate for each threshold of a method, which equals to Correction/Sum×100%.

As a whole, the precision of Overlap is still higher than that of the embedding methods, but it is easy for us to find out that the correlation number is limited, far less than that of word embedding methods. Obviously, content linking with word embedding models can rely on not only shallow semantics, but also deeper semantic understanding. For example, "Does anyone really think they will stop this now?" and "This was never about getting the establishment to change.", these sentences expressed that the situation will not change, which are not the same from the perspective of morphological view. In other words, word embedding models can really catch content linking results with deeper semantics connected. But

Table 6. Content linking results

Method	Threshold	Correction	Sum	Precision
Overlap	0.1	291	344	84.59%
	0.2	**199**	**226**	**88.05%**
	0.4	79	91	86.81%
GloVe	0.4	681	945	72.06%
	0.6	**495**	**679**	**72.90%**
	0.8	126	176	71.59%
Word2Vec	0.2	465	627	74.16%
	0.4	**366**	**479**	**76.41%**
	0.6	150	199	75.38%

the price is more loss in precision compared with Overlap. This is just the problem of "excessive linking" which we have mentioned before in section 4.1 for Chinese TianYa corpus. Hence we really need to work more to improve the precision through more precise word embedding models.

As to the thresholds, we can also see from Table 6 that different thresholds for different methods can really affect the accuracy of linking performance. Generally speaking, when the threshold is lower, we can extract more links, but there may be more wrong links selected at the same time, which may leads to precision loss. It occurs to us that there should be a best threshold in some middle point for different methods, although the concrete value of the threshold may vary for different methods.

There may be another reason for these results. The training corpus of our word vectors in this English experiment is collected from Wikipedia by us, which is different from the testing corpus of OnForumS from The Guardian. Usually we believe that different websites have different language styles, which may possibly affect the training result of the word embedding models. We will also study more about training word embedding models and hope to obtain better word vectors via more fitful training experiments.

5 Conclusion

In this paper, we mainly study the task of content linking between comment sentence and article sentence or former comment sentence for online forums. 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. Our experiments have been implemented for both English and Chinese realistic forum data. The results 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 for "excessive linking" via integration of various methods so as to make full use of merits of existed technologies and more precise word embedding models with good training.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant 71231002, 61202247, 61202248 and 61472046; EU FP7 IRSES Mobile Cloud Project (Grant No. 612212); the 111 Project of China under Grant B08004; Engineering Research Center of Information Networks, Ministry of Education.

References

[1] M. Kabadjov, U. Kruschwitz, M. Poesio, J. Steinberger, OnForumS: A shared task on on-line forum summarisation. 〈http://multiling.iit.demokritos.gr/pages/view/1573/multiling-2015-proceedings-addendum〉.

[2] A. Aker, E. Kurtic, M. Hepple, R. Gaizauskas, G.D. Fabbrizio, Comment-to-article linking in the online news domain, in: Proc. of the SIGDIAL 2015 Conference, 2015.

[3] H. Tanev, A. Balahur, Tackling the OnForumS challenge. 〈http://multiling.iit.demokritos.gr/pages/view/1573/multiling-2015-proceedings-addendum〉.

[4] A. Aker, F. Celli, A. Funk, E. Kurtic, M. Hepple, R. Gaizauskas, Sheffield-trento system for sentiment and argument structure enhanced comment-to-article linking in the online news domain. 〈http://multiling.iit.demokritos.gr/pages/view/1573/multiling-2015-proceedings-addendum〉.

[5] P. Krejzl, J. Steinberger, Tom´aˇsHercig, Tom´aˇsBrychcín, UWB participation in the multiling's OnForumS task. 〈http://multiling.iit.demokritos.gr/pages/view/1573/multiling-2015-proceedings-addendum〉.

[6] B. Pang, L. Lee, Opinion mining and sentiment analysis, Foundations and Trends in Information Retrieval 2(1-2)(2008) 1-135.

[7] S. Li, Q. Ye, Y. Li, R. Law, Mining features of product from Chinese customer online reviews, Journal of Management Sciences in China 12(2)(2009) 142-152.

[8] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313(5786)(2006) 504-507.

[9] A. Mnih, G. Hinton, Three new graphical models for statistical language modelling, in: Proc. of the 24th international conference on Machine learning ACM, 2007.

[10] E.H Huang, R. Socher, C.D. Manning, A.Y. Ng, Improving word representations via global context and multiple word prototypes, in: Proc. of the Annual Meeting of the Association for Computational Linguistics ACL, 2012.

[11] 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.

[12] R. Socher, J. Pennington, E.H. Huang, A.Y. Ng, , C.D. Manning, Semi-supervised recursive autoencoders for predicting sentiment distributions, in: Proc. of the Empiricial Methods in Natural Language Processing (EMNLP 2011), 2011.

[13] J. Turian, L. Ratinov, Y. Bengio, D. Roth, A preliminary evaluation of word representations for named-entity recognition, in: NIPS Workshop on Grammar Induction, Representation of Language and Language Learning, 2009.

[14] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space. 〈http://arxiv.org/abs/1301.3781v3, 2013〉.

[15] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Proc. of NIPS, 2013.

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

[17] A. Mnih, G.E. Hinton, A scalable hierarchical distributed language model, in: Proc. of NIPS, 2008.