A Fast and Effective Method for Clustering Large-Scale Chinese Question Dataset

Xiaodong Zhang and Houfeng Wang

Key Laboratory of Computational Linguistics, Peking University,
Ministry of Education, China
zxddavy@gmail.com, wanghf@pku.edu.cn

Abstract. Question clustering plays an important role in QA systems. Due to data sparseness and lexical gap in questions, there is no sufficient information to guarantee good clustering results. Besides, previous works pay little attention to the complexity of algorithms, resulting in infeasibility on large-scale datasets. In this paper, we propose a novel similarity measure, which employs word relatedness as additional information to help calculating similarity between questions. Based on the similarity measure and k-means algorithm, semantic k-means algorithm and its extended version are proposed. Experimental results show that the proposed methods have comparable performance with state-of-the-art methods and cost less time.

Keywords: Question Clustering, Word Relatedness, Semantic K-means.

1 Introduction

In recent years, short texts, such as snippets, micro-blogs, and questions etc., are prevalent on the Internet. Community Question Answering (CQA) websites have accumulated large archives of question-answer pairs, which promote the development of the question-answer datasets based QA system. Such QA systems retrieve questions from the dataset, which are semantically equivalent or relevant to queried questions, and show corresponding answers to users. To retrieve questions fast in a large-scale dataset, one feasible way is to cluster questions in advance so as to reduce the retrieval range.

There are two major challenges for question clustering. Firstly, question clustering faces data sparseness problem. Unlike normal texts with lots of words, questions only consist of several sentences (even just a few words). They do not provide sufficient statistical information, e.g. word co-occurrence, for effective similarity measure [1]. Secondly, question clustering suffers from lexical gap problem. Different words are used to express the same meaning in human languages. Conventional methods usually ignore the useful information of literally different but related words. In addition to these two challenges, the complexity of clustering algorithm should be concerned about. There are more than 300 million questions in zhidao[1], the most famous Chinese CQA, and the number keeps

[1] http://zhidao.baidu.com/

growing. Time-consuming or space-consuming methods cannot be performed on large-scale datasets without costly hardware resources. It is beneficial that the clustering algorithm is both effective and fast.

In order to settle the problems in question clustering, we first present a novel question similarity measure, which uses word relatedness to bridge the lexical gap between questions. The relation of two questions is modeled as a bipartite graph, based on which we define the similarity of two questions. Next we propose an effective and fast question clustering algorithm, referred to as semantic k-means (Sk-means), by introducing the proposed similarity measure into the k-means algorithm. An extended method, referred to as extended semantic k-means (ESk-means), is presented to improve the effectiveness further and gives more choices of the balance of effectiveness and complexity. We classified 16000 Chinese questions manually and built a classified question dataset to compare the performances of our method and some other popular approaches. The experimental results show that our proposed methods are very successful.

2 Related Works

Many methods have been proposed to improve short text clustering. Some researchers employed name entities [2] and phrases [3] extracted from the original text to construct the feature space, which is called surface representation. Ni [4] presented a novel clustering strategy, TermCut, which recursively select a core term and bisect text graph according to the term. Lack of semantic knowledge, these techniques suffer from the lexical gap problem. Another way is to enrich the text representation based on "bag of words" model by generating external features from linguistic and collaborative knowledge bases. Hotho [5] observed that additional features from WordNet can improve clustering results. Somnath [6] proposed a method to enrich short texts representation with additional features from Wikipedia. However, enriching the representation by knowledge usually require structured knowledge bases, e.g. WordNet and Wikipedia etc., which is scarce in Chinese circumstance.

Topic models have been proposed to uncover the latent semantic structure from text corpus and can be used for clustering. Latent Semantic Analysis (LSA) [7], Probabilistic Latent Semantic Analysis (PLSA) [8] and Latent Dirichlet Allocation (LDA) [9] have been proved successful on normal texts. Hong [10] made a comprehensive empirical study of topic modeling in Twitter, and suggested that new topic models for short texts are in demand. Yan [11] proposed a biterm topic model for short texts, which learn the topics by directly modeling the generation of word co-occurrence patterns. Ji [12] presented a Question-Answer Topic Model (QATM) to learn the latent topics aligned across the question-answer pairs to alleviate the lexical gap problem. Guo [13] proposed a Weighted Textual Matrix Factorization (WTMF) method to model missing words appropriately. The major challenge of topic models is that short texts do not provide sufficient word co-occurrence or other statistics to learn hidden variables. The performance of topic model in short text is not as good as in normal text.

3 Question Similarity Measure

Similarity or distance measure plays an important role in clustering algorithm based on text similarity. Compared to normal text, questions suffer from data sparseness and lexical gap. There are probably only a few (even none) words in common between two related questions. Vector space model (VSM) is the most commonly used text representation method. Based on VSM, it is inaccurate to calculate question similarity by conventional similarity measures, e.g. cosine similarity. Consider the two following Chinese questions:

Question 1: 电脑出故障，过了保修期怎么办？(My computer broke down and its warranty expired. What should I do?)

Question 2: 我想给笔记本装个固态硬盘，哪个牌子比较好？(I would like to install a SSD to my laptop. Which brand is good?)

Both about computers, the two questions are highly correlative. However, there are no words literally same between the two questions so that an extremely low correlation is given by cosine similarity. Cosine similarity considers that the relation between words is binary, literally same or not, ignoring different relatedness between words.

We introduce word relatedness as semantic information into our question similarity measure. Many methods were proposed for calculating word relatedness [14, 15]. We use word2vec[2] to calculate word relatedness in this work. This tool provides an efficient implementation of the continuous bag-of-words and skip-gram architectures [16, 17] for computing vector representations of words. Word relatedness can be defined as cosine similarity of the vector representations, as is shown in (1), where t_1 and t_2 are two words and v_1 and v_2 are their vector representations, respectively. If the training dataset is large enough, most words that do not appear in training dataset are noises and only a small part is neologisms, which have little influence and can be solved by updating training dataset. The major advantage of this method is that the training data only needs unstructured plain texts, which is easier to acquire than structured resources, e.g. semantic dictionary.

$$r\left(t_1, t_2\right) = \begin{cases} 1 & t_1 = t_2 \\ 0 & t_1 \text{ or } t_2 \text{ not in training data} \\ \dfrac{\langle v_1 \cdot v_2 \rangle}{\|v_1\| \, \|v_2\|} & \text{otherwise} \end{cases} \tag{1}$$

In our method, questions are represented by VSM. There are various choices for term weights in vectors. Here we use the popular TF-IDF weight and the vectors are normalized. Then we model the similarity of two questions by a bipartite graph. Let $G = (U, V, E)$ denote a bipartite graph whose partition has parts U and V, with E denoting the edges of the graph. Let q_1 and q_2 be two questions. The graph is constructed as follows. For each word type t_{1i} ($i \in [1, n_1]$, n_1 is the number of word types in q_1) in q_1, there is a corresponding node u_i in U. Node v_j in V for each word type t_{2j} in q_2 is defined in the same way. If the

[2] https://code.google.com/p/word2vec/

relatedness of t_{1i} and t_{2j} exceeds a threshold (the detail of threshold setting is discussed in Sect. 5), the two words are considered related and nodes u_i and v_j are connected by a edge, of which the weight is calculated as follows:

$$e_{ij} = w_{1i} \times w_{2j} \times r_{ij} \tag{2}$$

where e_{ij} is the weight of the edge connecting u_i and v_j, w_{1i} and w_{2j} are the TF-IDF weights of word t_{1i} and t_{2j} in vectors, r_{ij} is the relatedness of t_{1i} and t_{2j}. The similarity of two questions is defined as the sum of weights of edges in the maximum weight matching of the bipartite graph. Formally,

$$s = \sum_{e \in \mathrm{MWM}(G)} w_e \tag{3}$$

where MWM(G) are edges belong to the maximum weight matching of the bipartite graph G and w_e is the weight of edge e.

The maximum weight matching can be solved by Kuhn-Munkres algorithm, with $O(n^3)$ time complexity, where n is the number of nodes in the graph. Although the time complexity is high, the maximum weight matching can be calculated within a short time because n is small in the case of short text (the average number of words in a question is 12.5 in our dataset).

Next we give a concrete example of calculating the similarity of the two questions mentioned above. The vector representations of the two questions are as follows and the decimal after slash is word weight, which is calculated in a true dataset. Question 1: [电脑/0.44, 出/0.11, 故障/0.55, 过/0.17, 保修期/0.67, 怎么办/0.11]; Question 2: [笔记本/0.47, 装/0.27, 固态硬盘/0.80, 哪个/0.07, 牌子/0.33 好/0.07]. Pairs of words that relatedness exceeds 0.2 (threshold) are as follows: (电脑, 笔记本) = 0.78; (电脑, 固态硬盘) = 0.52; (电脑, 装) = 0.21; (故障, 笔记本) = 0.28; (故障, 固态硬盘) = 0.22; (保修期, 笔记本) = 0.28. Fig. 1 shows the bipartite graph of the two questions. The weights of edges (attached to lines) are computed by (2). The maximum weight matching is marked by bold lines and the similarity of the two questions is 0.26, rather than 0 computed by conventional similarity measures, e.g. cosine similarity.

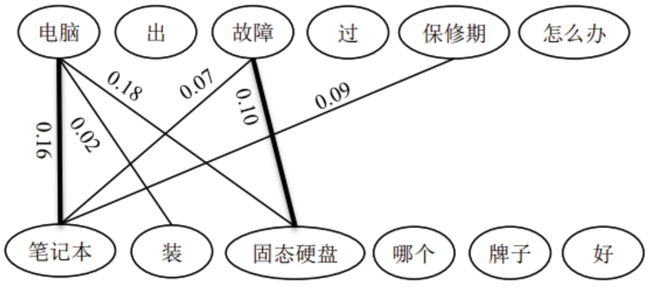

Fig. 1. Bipartite graph of two questions

Our method overcomes the two problems of calculating similarity of short texts. Word relatedness bridges the lexical gap between questions. Literally different but related words can contribute to the similarity calculation. With the help of additional word relatedness information, the sparse features of questions are enriched. Note that we treat questions as general short texts, so our proposed similarity measure can be used in other types of short texts, not just questions.

4 Question Clustering Method

4.1 Semantic K-means Algorithm

Based on our proposed question similarity measure and k-means algorithm, we present a question clustering method, referred to as semantic k-means (Sk-means). The method can also be used for clustering other types of short texts. First of all, we analyze the pros and cons of k-means algorithm in the case of question clustering so that we can present our method naturally.

The major advantage of k-means is low time complexity and space complexity, which is extremely useful for large-scale datasets. However, k-means has poor performance on short texts because it cannot solve the data sparseness and lexical gap problem in questions. In k-means algorithm, k data points are randomly chosen as the initial centroids. In the iterations, every data point is computed the distance to each centroid and then assigned to the closed centroid. Note that in the first iteration the centroid vectors are extremely sparse, which means a data point is hard to decide which centroid is closest to it by conventional similarity measures. A large number of wrong assignments will affect following iterations and eventual results. In the following iterations, however, the centroid vectors become much denser because of re-computing centroids by averaging all data points assigned to the centroids and the similarity calculation is more accurate than in the first iteration. Therefore, the major problem of k-means is at the first iteration. Based on the analysis above, an idea about improving k-means comes up intuitively. Our proposed similarity measure, which employs word relatedness as semantic information to improve question similarity calculation, is used in the first iteration so as to avoid the inaccurate similarity between sparse questions. The cosine similarity is used in the rest iterations to keep the high speed of the algorithm. Algorithm 1 shows the pseudo-code of Sk-means. The questions are represented by VSM so that each question can be regarded as a data point.

The main difference between Sk-means and k-means is that in the first iteration our method employs semantic information (word relatedness) to help similarity calculation of short texts, alleviating the data sparseness and lexical gap. Any stopping criterion used in k-means can be used in our method. In our experiments, the algorithm stops when no data point is assigned to different clusters between two consecutive iterations.

Algorithm 1. semantic k-means (k, D)

```
 1: choose $k$ data points randomly as the initial centroids (cluster centers);
 2: repeat
 3:    for each data point $x \in D$ do
 4:       if in the first iteration then
 5:          compute the similarity of $x$ and each centroid by our proposed similarity
            measure;
 6:       else
 7:          compute the similarity of $x$ and each centroid by cosine similarity
 8:       end if
 9:       assign $x$ to the most similar centroid
10:    end for
11:    re-compute the centroid using the current cluster memberships
12: until the stopping criterion is met
```

4.2 Extended Semantic K-means Algorithm

A confusing thing of our algorithm is why the proposed similarity measure is only used in the first iteration. This is mainly because of the balance of effectiveness and efficiency. The proposed similarity measure does help in the following iterations but is not as helpful as in the first iteration. As the centroids become dense in the following iterations, calculating the similarity between data points and centroids by our proposed approach is very time consuming. Therefore, the Sk-means algorithm uses our proposed measure only in the first iteration. Nevertheless, our proposed similarity measure can still help alleviating lexical gap in the following iterations. Thus we give an extended version of Sk-means, referred to as extended semantic k-means (ESk-means), to improve the effectiveness further at cost of higher complexity. The algorithm is shown in Algorithm 2. ESk-means uses our proposed similarity measure in the headmost m iterations, rather than only in the first iteration. A trick is used to reduce time consuming calculations. In the headmost m iterations, we truncate the centroids vectors and only reserve d dimensions with highest weights (all other dimensions are set to 0). Then the truncated vectors are normalized and used for similarity calculation by our proposed measure. In the following iterations, cosine similarity is used and the centroid vectors are not truncated. We can see that ESk-means is just Sk-means if $m = 1$ and the truncation is not performed.

4.3 Complexity Analysis

Next we compare the time and space complexity of our proposed clustering methods with some other clustering methods. The comparisons are shown in Table 1, including six clustering methods, in which BTM [11] is a state-of-the-art topic model for short texts. The notation t is the number of iterations, n is the number of document in dataset, k is the number of clusters, and $\bar{l}$ is average length of a document. In ESk-means, m is the number of iterations using the proposed similarity measure, d is the number of reserved dimensions. In BTM, b

Algorithm 2. extended semantic k-means (k, D, m, d)

```
1: choose $k$ data points randomly as the initial centroids (cluster centers);
2: $iter \leftarrow 1$
3: repeat
4:     truncate and normalize centroids (reserve $d$ dimensions)
5:     for each data point $x \in D$ do
6:         compute the similarity of $x$ and each centroid by our proposed similarity measure;
7:         assign $x$ to the most similar centroid
8:     end for
9:     re-compute the centroid using the current cluster memberships
10:    $iter \leftarrow iter + 1$
11: until $iter > m$
12: repeat
13:    for each data point $x \in D$ do
14:        compute the similarity of $x$ and each centroid by cosine similarity;
15:        assign $x$ to the most similar centroid
16:    end for
17:    re-compute the centroid using the current cluster memberships
18: until the stopping criterion is met
```

is the number of biterms and can be approximately rewritten as $b \approx (n\bar{l}(\bar{l}-1))/2$, and v is the number of word types. Note that different methods have different number of iterations, which is distinguished by subscripts. Usually t_4 and t_5 are much larger than t_1, t_2, and t_3. K-means and our methods usually take tens of rounds before stop, while LDA and BTM take hundreds and even thousands of rounds before stop. $\bar{l}$ is small for questions and can be considered as constant. From the comparisons, we can see that Sk-means has very close time and space complexity with k-means. The complexity of Sk-means is lower than LDA and BTM and the complexity of ESk-means depends on parameter m and d. The complexity of spectral clustering is too high to be used in large-scale datasets. In Sect. 5, we will give the real time consumption and the effectiveness of each method on a dataset.

Table 1. The complexity comparison of proposed methods and some other methods

Method	Time complexity	Space complexity
k-means	$O(t_1 nk\bar{l})$	$O((k+n)\bar{l})$
Sk-means	$O(nk\bar{l}^3 + t_2 nk\bar{l})$	$O((k+n)\bar{l} + \bar{l}^2)$
ESk-means	$O(mnkd^3 + t_3 nk\bar{l})$	$O(n\bar{l} + kd + d^2)$
LDA	$O(t_4 nk\bar{l})$	$O(nk + vk + n\bar{l})$
BTM	$O(t_5 kb)$	$O(k + vk + b)$
spectral clustering	$O(n^2 k)$	$O(n^2)$

5 Experiments

5.1 Corpus and Evaluation Metrics

As far as we know, there is no available large-scale classified Chinese question dataset. So we build a classified Chinese question dataset by collecting questions from zhidao[3]. Although the questions are classified by users when posting these questions, there is much misclassification for some reasons. We reviewed manually and get rid of the misclassified questions. The question dataset contains 16000 Chinese questions, which are classified into 8 classes and each class consists of 2000 questions. We segment each Chinese question using ICTCLAS[4] and remove the stop words by checking a stop word list containing 746 Chinese words. Each resultant Chinese word is used as term for further computation.

As for the training data of word2vec, we use 3 datasets acquired from the Internet, including Chinese Wikipedia[5], Chinese Gigaword[6], Sougou news corpus[7]. We remove labels and links from the 3 corpus and then segment the remaining text by ICTCLAS. The combination of the 3 corpus is used as training data for word2vec, which calculates the word relatedness.

The *FScore* measure is one of the commonest evaluation metrics for clustering task. We use *Micro Averaged FScore* [4] (denoted as *MicroFScore*) to test the effectiveness of the proposed methods. Due to space limitations, we do not give the definition in this paper.

5.2 Experiments Settings and Results Analysis

We carry out 8 experiments to compare the performance of the proposed methods with other popular methods. The settings are as follows:

Experiment 1: K-means algorithm is carried out, with cosine similarity as similarity measure.

Experiment 2: We perform spectral clustering, which is usually considered a better algorithm than k-means. Cosine similarity is used as similarity measure.

Experiment 3: This experiment is also spectral clustering. The difference from Experiment 2 is that our proposed similarity measure is used. We refer to this experiment as spectral++.

Experiment 4: We enrich the representation of questions with additional features from Wikipedia by the method proposed in [6]. Then k-means is carried out on the enriched representations. This experiment is referred to as wiki.

Experiment 5: LDA is carried out and the hyper parameters are tuned via grid search. In this experiment, $\alpha = 0.1$ and $\beta = 0.05$.

[3] `http://zhidao.baidu.com/`
[4] `http://ictclas.nlpir.org/`
[5] `http://download.wikipedia.com/zhwiki/`
[6] `https://catalog.ldc.upenn.edu/LDC2011T13`
[7] `http://www.sogou.com/labs/dl/ca.html`

Experiment 6: We perform BTM, which is state-of-the-art topic model for short texts. Parameters are also tuned via grid search. In this experiment, $\alpha = 0.1$ and $\beta = 0.01$.

Experiment 7: Sematic k-means algorithm is performed. The threshold in similarity measure is set to 0.3.

Experiment 8: We carry out extended semantic k-means algorithm. The threshold in similarity measure is set to 0.3. In this experiment, $m = 8$ and $d = 50$.

For each experiment, the number of cluster is set to 8, the actual class number. Because of the stochasticity of the algorithms, we perform each algorithm ten times and take the average score of ten times as final score. The consumed time is also the average time of ten times. The performance comparisons are shown in Table 2. The average score, highest score in 10 times and average consumed time are listed in the table.

Table 2. Comparisons of different methods

#	Method	MicroFScore (average)	MicroFScore (highest)	Time
1	k-means	0.644	0.751	6s
2	spectral	0.554	0.575	919s
3	spectral++	0.671	0.721	1725s
4	wiki	0.678	0.757	207s
5	LDA	0.734	0.798	32s
6	BTM	0.741	0.804	148s
7	Sk-means	0.736	0.821	10s
8	ESk-means	0.740	0.821	88s

From the experimental results, we can see that k-means is the fastest method among these methods but the effectiveness is unsatisfying. It is surprising that the score of spectral clustering is even lower than k-means. This is because it is inaccurate to calculate the similarity of questions by conventional similarity measures, resulting in unreliable similarity matrix and eigenvectors. In the iterations of k-means, however, the centroids become dense so that the similarity calculations become accurate. In Experiment 3, the similarity matrix is calculated by our proposed measure and then spectral clustering is performed. Based on this similarity matrix, the result of spectral clustering is better than Experiment 1 and 2, proving that our proposed similarity measure is effective. In Experiment 4, the f-score is 1.4% higher than k-means, showing that enrich the representation by Wikipedia is also helpful. However, the improvement of Experiment 3 and 4 is limited, only slightly higher than Experiment 1, and cost a lot of time. In Experiment 5 and 6, we find that the results of topic model are much better than k-means. In particular, the BTM achieves the highest performance among all experiments. We test the performance of our methods in Experiment 7 and 8. In Experiment 7, the Sk-means outperforms k-means, spectral clustering, wiki and LDA and the performance is only slightly lower than BTM. The speed of Sk-means is close to k-means. It is amazing that a method with low time complexity can have such good results. In Experiment 8, ESk-means takes more time to gain better results than Sk-means. The performance of ESk-means is comparable to BTM. Although the average score is slightly lower than BTM, the highest score in ten times is achieved by Sk-means and ESk-means. The performance of our proposed methods depends heavily on the initial centroids. For some good initial centroids, our methods can get extremely good results. Unfortunately, we have not found effective methods for choosing good initial centroids. As Sk-means costs little time, if there is way to evaluate (maybe approximately) results, we can run Sk-means many times to get a good result. From all these experiments, we can conclude that Sk-means is a fast and effective method for clustering questions and ESk-means can improve the results further at cost of more time.

The remaining unsettled thing is the setting of some parameters in our methods. First, we discuss the setting of the threshold in our proposed similarity measure. The threshold is used to determine whether two words are considered related. Here we examine the influence of the threshold via Sk-means. We run the algorithm 11 times on the dataset. The threshold is varied from 0 to 1 increased by 0.1 each time. The initial seeds are same in the 11 times. The results are shown in Fig. 2. When the threshold is set around 0.2 and 0.3, the result is good. When the threshold is lower, the information of unrelated words is introduced into the similarity measure and has negative effects. When the threshold is higher, less word relatedness information can be used so that the f-score decreases. As the threshold draws near 1, the Sk-means just becomes k-means actually. Therefore, 0.2 and 0.3 is the recommend threshold value.

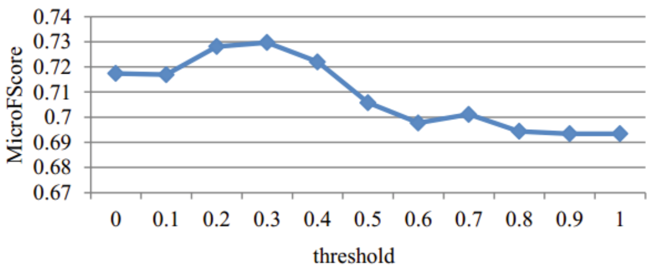

Fig. 2. MicroFscore changes when the threshold is set from 0 to 1

Next we discuss the setting of parameter m and d in ESk-means. An intuitive thought is that larger m and d will produce better results. As shown in Fig. 3, this thought is true within some limits. However, larger m and d mean more consumed time and space. We find that the improvement of f-score is not significant when

m is larger than 8 and d is larger than 50. Therefore, a reasonable choice is setting m to 8 and d to 50. Note that when d is small (e.g. 10), larger m makes the results worse. This is because the centroids consist of too little terms and the truncation makes the centroids lose too much useful information.

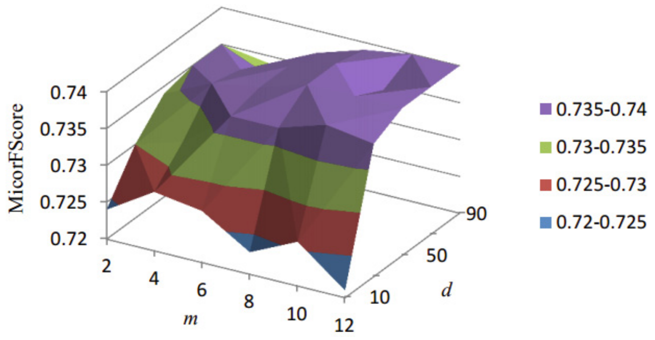

Fig. 3. The influence of parameter m and d in ESk-means

6 Conclusion and Future Work

The CQA websites have accumulated quantities of questions. However, there is no fast and effective clustering method for these large-scale datasets. Our work mainly consists of two parts. Firstly, we propose a novel similarity measure for questions. Word relatedness is employed to tackle the problems of data sparseness and lexical gap in questions. The relation of two questions is modeled by bipartite graph, based on which we define the similarity of two questions. Secondly, we propose Sk-means algorithm and ESk-means algorithm by introducing our proposed similarity measure into k-means algorithm. The experimental results show that Sk-means is a fast and effective method for clustering questions and ESk-means can improve the results further at cost of more time.

There are some interesting future works to be continued. We will explore the application of our similarity measure in other clustering methods and NLP tasks. In consideration of the good results of topic model in the experiments, we will try to introduce word relatedness into topic model to improve question clustering.

Acknowledgments. This research was partly supported by National Natural Science Foundation of China (No. 61370117, 61333018), Major National Social Science Fund of China(No.12&ZD227), National High Technology Research and Development Program of China (863 Program) (No. 2012AA011101).

References

1. Phan, X.H., Nguyen, L.M., Horiguchi, S.: Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In: Proceedings of the 17th International Conference on World Wide Web, pp. 91–100. ACM (April 2008)
2. Kumaran, G., Allan, J.: Text classification and named entities for new event detection. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 297–304. ACM (July 2004)
3. Chim, H., Deng, X.: Efficient phrase-based document similarity for clustering. IEEE Transactions on Knowledge and Data Engineering 20(9), 1217–1229 (2008)
4. Ni, X., Quan, X., Lu, Z., Wenyin, L., Hua, B.: Short text clustering by finding core terms. Knowledge and Information Systems 27(3), 345–365 (2011)
5. Hotho, A., Staab, S., Stumme, G.: Ontologies improve text document clustering. In: Third IEEE International Conference on Data Mining, ICDM 2003, pp. 541–544. IEEE (November 2003)
6. Banerjee, S., Ramanathan, K., Gupta, A.: Clustering short texts using wikipedia. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 787–788. ACM (July 2007)
7. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. JASIS 41(6), 391–407 (1990)
8. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 50–57. ACM (August 1999)
9. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. The Journal of Machine Learning Research 3, 993–1022 (2003)
10. Hong, L., Davison, B.D.: Empirical study of topic modeling in twitter. In: Proceedings of the First Workshop on Social Media Analytics, pp. 80–88. ACM (July 2010)
11. Yan, X., Guo, J., Lan, Y., Cheng, X.: A biterm topic model for short texts. In: Proceedings of the 22nd International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, pp. 1445–1456 (May 2013)
12. Ji, Z., Xu, F., Wang, B., He, B.: Question-answer topic model for question retrieval in community question answering. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 2471–2474. ACM (October 2012)
13. Guo, W., Diab, M.: Modeling sentences in the latent space. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers, vol. 1, pp. 864–872. Association for Computational Linguistics (July 2012)
14. Strube, M., Ponzetto, S.P.: WikiRelate! Computing semantic relatedness using Wikipedia. In: AAAI, vol. 6, pp. 1419–1424 (July 2006)
15. Gracia, J.L., Mena, E.: Web-based measure of semantic relatedness. In: Bailey, J., Maier, D., Schewe, K.-D., Thalheim, B., Wang, X.S. (eds.) WISE 2008. LNCS, vol. 5175, pp. 136–150. Springer, Heidelberg (2008)
16. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)