Modeling Semantic Relevance for Question-Answer Pairs in Web Social Communities

Baoxun Wang, Xiaolong Wang, Chengjie Sun, Bingquan Liu, Lin Sun
School of Computer Science and Technology
Harbin Institute of Technology
Harbin, China
{bxwang, wangxl, cjsun, liubq, lsun}@insun.hit.edu.cn

Abstract

Quantifying the semantic relevance between questions and their candidate answers is essential to answer detection in social media corpora. In this paper, a deep belief network is proposed to model the semantic relevance for question-answer pairs. Observing the textual similarity between the community-driven question-answering (cQA) dataset and the forum dataset, we present a novel learning strategy to promote the performance of our method on the social community datasets without hand-annotating work. The experimental results show that our method outperforms the traditional approaches on both the cQA and the forum corpora.

1 Introduction

In natural language processing (NLP) and information retrieval (IR) fields, question answering (QA) problem has attracted much attention over the past few years. Nevertheless, most of the QA researches mainly focus on locating the exact answer to a given factoid question in the related documents. The most well known international evaluation on the factoid QA task is the Text REtrieval Conference (TREC)[1], and the annotated questions and answers released by TREC have become important resources for the researchers. However, when facing a non-factoid question such as *why*, *how*, or *what about*, however, almost no automatic QA systems work very well.

The user-generated question-answer pairs are definitely of great importance to solve the non-factoid questions. Obviously, these natural QA pairs are usually created during people's communication via Internet social media, among which we are interested in the community-driven question-answering (cQA) sites and online forums. The cQA sites (or systems) provide platforms where users can either ask questions or deliver answers, and best answers are selected manually (e.g., Baidu Zhidao[2] and Yahoo! Answers[3]). Comparing with cQA sites, online forums have more virtual society characteristics, where people hold discussions in certain domains, such as techniques, travel, sports, etc. Online forums contain a huge number of QA pairs, and much noise information is involved.

To make use of the QA pairs in cQA sites and online forums, one has to face the challenging problem of distinguishing the questions and their answers from the noise. According to our investigation, the data in the community based sites, especially for the forums, have two obvious characteristics: (a) a post usually includes a very short content, and when a person is initializing or replying a post, an informal tone tends to be used; (b) most of the posts are useless, which makes the community become a noisy environment for question-answer detection.

In this paper, a novel approach for modeling the semantic relevance for QA pairs in the social media sites is proposed. We concentrate on the following two problems:

1. How to model the semantic relationship between two short texts using simple textual features? As mentioned above, the user generated questions and their answers via social media are always short texts. The limitation of length leads to the sparsity of the word features. In addition, the word frequency is usually either 0 or 1, that is, the frequency offers little information except the occurrence of a word. Because of this situation, the traditional relevance computing methods based on word co-occurrence, such as Cosine similarity and KL-divergence, are not effective for question-

[1] http://trec.nist.gov

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

[3] http://answers.yahoo.com

Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 1230–1238,
Uppsala, Sweden, 11-16 July 2010.

answer semantic modeling. Most researchers try to introduce structural features or users' behavior to improve the models performance, by contrast, the effect of textual features is not obvious.

2. How to train a model so that it has good performance on both cQA and forum datasets? So far, people have been doing QA researches on the cQA and the forum datasets separately (Ding et al., 2008; Surdeanu et al., 2008), and no one has noticed the relationship between the two kinds of data. Since both the cQA systems and the online forums are open platforms for people to communicate, the QA pairs in the cQA systems have similarity with those in the forums. In this case, it is highly valuable and desirable to propose a training strategy to improve the model's performance on both of the two kinds of datasets. In addition, it is possible to avoid the expensive and arduous hand-annotating work by introducing the method.

To solve the first problem, we present a deep belief network (DBN) to model the semantic relevance between questions and their answers. The network establishes the semantic relationship for QA pairs by minimizing the answer-to-question reconstructing error. Using only word features, our model outperforms the traditional methods on question-answer relevance calculating.

For the second problem, we make our model to learn the semantic knowledge from the solved question threads in the cQA system. Instead of mining the structure based features from cQA pages and forum threads individually, we consider the textual similarity between the two kinds of data. The semantic information learned from cQA corpus is helpful to detect answers in forums, which makes our model show good performance on social media corpora. Thanks to the labels for the best answers existing in the threads, no manual work is needed in our strategy.

The rest of this paper is organized as follows: Section 2 surveys the related work. Section 3 introduces the deep belief network for answer detection. In Section 4, the homogenous data based learning strategy is described. Experimental result is given in Section 5. Finally, conclusions and future directions are drawn in Section 6.

2 Related Work

The value of the naturally generated question-answer pairs has not been recognized until recent years. Early studies mainly focus on extracting QA pairs from frequently asked questions (FAQ) pages (Jijkoun and de Rijke, 2005; Riezler et al., 2007) or service call-center dialogues (Berger et al., 2000).

Judging whether a candidate answer is semantically related to the question in the cQA page automatically is a challenging task. A framework for predicting the quality of answers has been presented in (Jeon et al., 2006). Bernhard and Gurevych (2009) have developed a translation based method to find answers. Surdeanu et al. (2008) propose an approach to rank the answers retrieved by Yahoo! Answers. Our work is partly similar to Surdeanu et al. (2008), for we also aim to rank the candidate answers reasonably, but our ranking algorithm needs only word information, instead of the combination of different kinds of features.

Because people have considerable freedom to post on forums, there are a great number of irrelevant posts for answering questions, which makes it more difficult to detect answers in the forums. In this field, exploratory studies have been done by Feng et al. (2006) and Huang et al. (2007), who extract input-reply pairs for the discussion-bot. Ding et al.(2008) and Cong et al.(2008) have also presented outstanding research works on forum QA extraction. Ding et al. (2008) detect question contexts and answers using the conditional random fields, and a ranking algorithm based on the authority of forum users is proposed by Cong et al. (2008). Treating answer detection as a binary classification problem is an intuitive idea, thus there are some studies trying to solve it from this view (Hong and Davison, 2009; Wang et al., 2009). Especially Hong and Davison (2009) have achieved a rather high precision on the corpora with less noise, which also shows the importance of "social" features.

In order to select the answers for a given question, one has to face the problem of lexical gap. One of the problems with lexical gap embedding is to find similar questions in QA achieves (Jeon et al., 2005). Recently, the statistical machine translation (SMT) strategy has become popular. Lee et al. (2008) use translate models to bridge the lexical gap between queries and questions in QA collections. The SMT based methods are effective on modeling the semantic relationship between questions and answers and expending users' queries in answer retrieval (Riezler et al., 2007; Berger et al.,

2000; Bernhard and Gurevych, 2009). In (Surdeanu et al., 2008), the translation model is used to provide features for answer ranking.

The structural features (e.g., authorship, acknowledgement, post position, etc), also called non-textual features, play an important role in answer extraction. Such features are used in (Ding et al., 2008; Cong et al., 2008), and have significantly improved the performance. The studies of Jeon et al. (2006) and Hong et al. (2009) show that the structural features have even more contribution than the textual features. In this case, the mining of textual features tends to be ignored.

There are also some other research topics in this field. Cong et al. (2008) and Wang et al. (2009) both propose the strategies to detect questions in the social media corpus, which is proved to be a non-trivial task. The deep research on question detection has been taken by Duan et al. (2008). A graph based algorithm is presented to answer opinion questions (Li et al., 2009). In email summarization field, the QA pairs are also extracted from email contents as the main elements of email summarization (Shrestha and McKeown, 2004).

3 The Deep Belief Network for QA pairs

Due to the feature sparsity and the low word frequency of the social media corpus, it is difficult to model the semantic relevance between questions and answers using only co-occurrence features. It is clear that the semantic link exists between the question and its answers, even though they have totally different lexical representations. Thus a specially designed model may learn semantic knowledge by reconstructing a great number of questions using the information in the corresponding answers. In this section, we propose a deep belief network for modeling the semantic relationship between questions and their answers. Our model is able to map the QA data into a low-dimensional semantic-feature space, where a question is close to its answers.

3.1 The Restricted Boltzmann Machine

An ensemble of binary vectors can be modeled using a two-layer network called a "restricted Boltzmann machine" (RBM) (Hinton, 2002). The dimension reducing approach based on RBM initially shows good performance on image processing (Hinton and Salakhutdinov, 2006). Salakhutdinov and Hinton (2009) propose a deep graphical model composed of RBMs into the information retrieval field, which shows that this model is able to obtain semantic information hidden in the word-count vectors.

As shown in Figure 1, the RBM is a two-layer network. The bottom layer represents a visible vector **v** and the top layer represents a latent feature **h**. The matrix W contains the symmetric interaction terms between the visible units and the hidden units. Given an input vector **v**, the trained

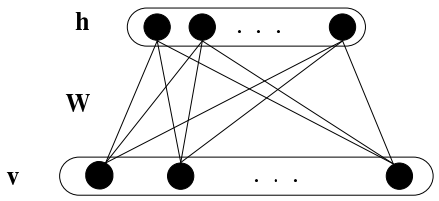

Figure 1: Restricted Boltzmann machine

RBM model provides a hidden feature **h**, which can be used to reconstruct **v** with a minimum error. The training algorithm for this paper will be described in the next subsection. The ability of the RBM suggests us to build a deep belief network based on RBM so that the semantic relevance between questions and answers can be modeled.

3.2 Pretraining a Deep Belief Network

In the social media corpora, the answers are always descriptive, containing one or several sentences. Noticing that an answer has strong semantic association with the question and involves more information than the question, we propose to train a deep belief network by reconstructing the question using its answers. The training object is to minimize the error of reconstruction, and after the pretraining process, a point that lies in a good region of parameter space can be achieved.

Firstly, the illustration of the DBN model is given in Figure 2. This model is composed of three layers, and here each layer stands for the RBM or its variant. The bottom layer is a variant form of RBM's designed for the QA pairs. This layer we design is a little different from the classical RBM's, so that the bottom layer can generate the hidden features according to the visible answer vector and reconstruct the question vector using the hidden features. The pre-training procedure of this architecture is practically convergent. In the bottom layer, the binary feature vectors based on the statistics of the word occurrence in the answers are used to compute the "hidden features" in the

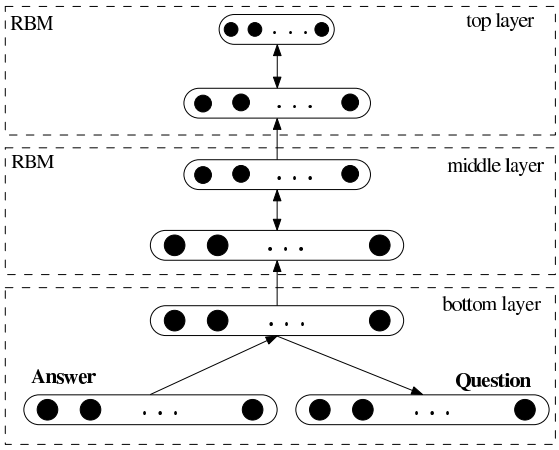

Figure 2: The Deep Belief Network for QA Pairs

hidden units. The model can reconstruct the questions using the hidden features. The processes can be modeled as follows:

$$p(h_j = 1|\mathbf{a}) = \sigma(b_j + \sum_i w_{ij} a_i) \quad (1)$$

$$p(q_i = 1|\mathbf{h}) = \sigma(b_i + \sum_j w_{ij} h_j) \quad (2)$$

where $\sigma(x) = 1/(1 + e^{-x})$, $\mathbf{a}$ denotes the visible feature vector of the answer, q_i is the *ith* element of the question vector, and $\mathbf{h}$ stands for the hidden feature vector for reconstructing the questions. w_{ij} is a symmetric interaction term between word i and hidden feature j, b_i stands for the bias of the model for word i, and b_j denotes the bias of hidden feature j.

Given the training set of answer vectors, the bottom layer generates the corresponding hidden features using Equation 1. Equation 2 is used to reconstruct the Bernoulli rates for each word in the question vectors after stochastically activating the hidden features. Then Equation 1 is taken again to make the hidden features active. We use 1-step Contrastive Divergence (Hinton, 2002) to update the parameters by performing gradient ascent:

$$\Delta w_{ij} = \epsilon(< q_i h_j >_{qData} - < q_i h_j >_{qRecon}) \quad (3)$$

where $< q_i h_j >_{qData}$ denotes the expectation of the frequency with which the word i in a question and the feature j are on together when the hidden features are driven by the question data. $< q_i h_j >_{qRecon}$ defines the corresponding expectation when the hidden features are driven by the reconstructed question data. ϵ is the learning rate.

The classical RBM structure is taken to build the middle layer and the top layer of the network. The training method for the higher two layer is similar to that of the bottom one, and we only have to make each RBM to reconstruct the input data using its hidden features. The parameter updates still obeying the rule defined by gradient ascent, which is quite similar to Equation 3. After training one layer, the $\mathbf{h}$ vectors are then sent to the higher-level layer as its "training data".

3.3 Fine-tuning the Weights

Notice that a greedy strategy is taken to train each layer individually during the pre-training procedure, it is necessary to fine-tune the weights of the entire network for optimal reconstruction. To fine-tune the weights, the network is unrolled, taking the answers as the input data to generate the corresponding questions at the output units. Using the cross-entropy error function, we can then tune the network by performing backpropagation through it. The experiment results in section 5.2 will show fine-tuning makes the network performs better for answer detection.

3.4 Best answer detection

After pre-training and fine-tuning, a deep belief network for QA pairs is established. To detect the best answer to a given question, we just have to send the vectors of the question and its candidate answers into the input units of the network and perform a level-by-level calculation to obtain the corresponding feature vectors. Then we calculate the distance between the mapped question vector and each candidate answer vector. We consider the candidate answer with the smallest distance as the best one.

4 Learning with Homogenous Data

In this section, we propose our strategy to make our DBN model to detect answers in both cQA and forum datasets, while the existing studies focus on one single dataset.

4.1 Homogenous QA Corpora from Different Sources

Our motivation of finding the homogenous question-answer corpora from different kind of social media is to guarantee the model's performance and avoid hand-annotating work.

In this paper, we get the "solved question" pages in the computer technology domain from Baidu Zhidao as the cQA corpus, and the threads of

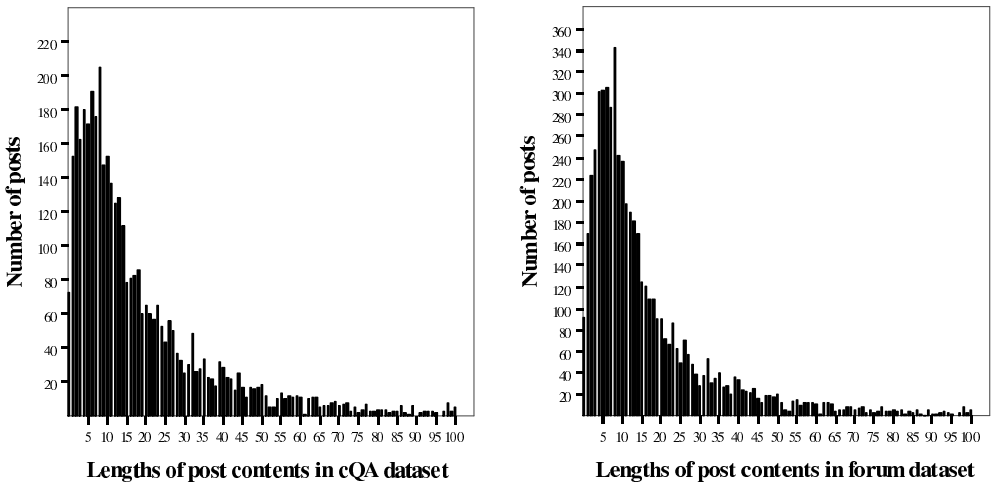

Figure 3: Comparison of the post content lengths in the cQA and the forum datasets

ComputerFansClub Forum[4] as the online forum corpus. The domains of the corpora are the same. To further explain that the two corpora are homogenous, we will give the detail comparison on text style and word distribution.

As shown in Figure 3, we have compared the post content lengths of the cQA and the forum in our corpora. For the comparison, 5,000 posts from the cQA corpus and 5,000 posts from the forum corpus are randomly selected. The left panel shows the statistical result on the Baidu Zhidao data, and the right panel shows the one on the forum data. The number i on the horizontal axis denotes the post contents whose lengths range from $10(i-1)+1$ to $10i$ bytes, and the vertical axis represents the counts of the post contents. From Figure 3 we observe that the contents of most posts in both the cQA corpus and the forum corpus are short, with the lengths not exceeding 400 bytes.

The content length reflects the text style of the posts in cQA systems and online forums. From Figure 3 it can be also seen that the distributions of the content lengths in the two figures are very similar. It shows that the contents in the two corpora are both mainly short texts.

Figure 4 shows the percentage of the concurrent words in the top-ranked content words with high frequency. In detail, we firstly rank the words by frequency in the two corpora. The words are chosen based on a professional dictionary to guarantee that they are meaningful in the computer knowledge field. The number k on the horizontal axis in Figure 4 represents the top k content words in the corpora, and the vertical axis stands for the percentage of the words shared by the two corpora in the top k words.

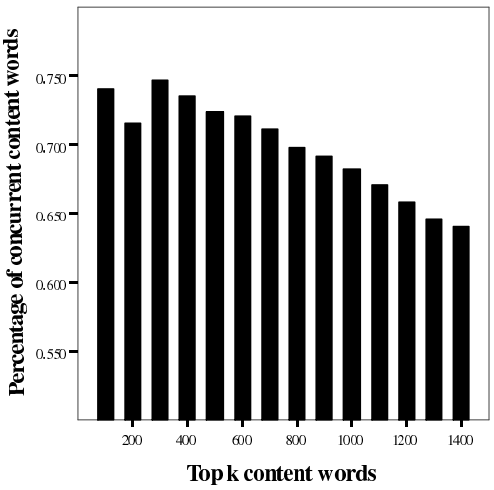

Figure 4: Distribution of concurrent content words

Figure 4 shows that a large number of meaningful words appear in both of the two corpora with high frequencies. The percentage of the concurrent words maintains above 64% in the top 1,400 words. It indicates that the word distributions of the two corpora are quite similar, although they come from different social media sites.

Because the cQA corpus and the forum corpus used in this study have homogenous characteristics for answer detecting task, a simple strategy may be used to avoid the hand-annotating work. Apparently, in every "solved question" page of Baidu Zhidao, the best answer is selected by the user who asks this question. We can easily extract the QA pairs from the cQA corpus as the training

[4]http://bbs.cfanclub.net/

set. Because the two corpora are similar, we can apply the deep belief network trained by the cQA corpus to detect answers on both the cQA data and the forum data.

4.2 Features

The task of detecting answers in social media corpora suffers from the problem of feature sparsity seriously. High-dimensional feature vectors with only several non-zero dimensions bring large time consumption to our model. Thus it is necessary to reduce the dimension of the feature vectors.

In this paper, we adopt two kinds of word features. Firstly, we consider the 1,300 most frequent words in the training set as Salakhutdinov and Hinton (2009) did. According to our statistics, the frequencies of the rest words are all less then 10, which are not statistically significant and may introduce much noise.

We take the occurrence of some function words as another kind of features. The function words are quite meaningful for judging whether a short text is an answer or not, especially for the non-factoid questions. For example, in the answers to the causation questions, the words such as *because* and *so* are more likely to appear; and the words such as *firstly*, *then*, and *should* may suggest the answers to the manner questions. We give an example for function word selection in Figure 5.

> ***Q***: 我的电脑开机一直黑屏，无法进 bios，应该怎么办？(The screen of my computer turns black after booting, and I can not enter bios, what should I do?)
> ***A***: ***可以先***拔掉硬盘，试试开机，***如果***能开机，***然后***在 bios 里把光驱设置成第一启动，重装系统吧。 (You ***may firstly*** unplug the hard disc and try to boot, ***if*** it works, ***then*** you ***should*** set your CD-Rom as the boot driver in bios and reinstall the system.)

Figure 5: An example for function word selection

For this reason, we collect 200 most frequent function words in the answers of the training set. Then for every short text, either a question or an answer, a 1,500-dimensional vector can be generated. Specifically, all the features we have adopted are binary, for they only have to denote whether the corresponding word appears in the text or not.

5 Experiments

To evaluate our question-answer semantic relevance computing method, we compare our approach with the popular methods on the answer detecting task.

5.1 Experiment Setup

Architecture of the Network: To build the deep belief network, we use a 1500-1500-1000-600 architecture, which means the three layers of the network have individually 1,500×1,500, 1,500×1,000 and 1,000×600 units. Using the network, a 1,500-dimensional binary vector is finally mapped to a 600-dimensional real-value vector.

During the pretraining stage, the bottom layer is greedily pretrained for 200 passes through the entire training set, and each of the rest two layers is greedily pretrained for 50 passes. For fine-tuning we apply the method of conjugate gradients[5], with three line searches performed in each pass. This algorithm is performed for 50 passes to fine-tune the network.

Dataset: we have crawled 20,000 pages of "solved question" from the *computer and network* category of Baidu Zhidao as the cQA corpus. Correspondingly we obtain 90,000 threads from ComputerFansClub, which is an online forum on computer knowledge. We take the forum threads as our forum corpus.

From the cQA corpus, we extract 12,600 human generated QA pairs as the training set without any manual work to label the best answers. We get the contents from another 2,000 cQA pages to form a testing set, each content of which includes one question and 4.5 candidate answers on average, with one best answer among them. To get another testing dataset, we randomly select 2,000 threads from the forum corpus. For this training set, human work are necessary to label the best answers in the posts of the threads. There are 7 posts included in each thread on average, among which one question and at least one answer exist.

Baseline: To show the performance of our method, three main popular relevance computing methods for ranking candidate answers are considered as our baselines. We will briefly introduce them:

Cosine Similarity. Given a question $\mathbf{q}$ and its candidate answer $\mathbf{a}$, their cosine similarity can be computed as follows:

$$\cos(\mathbf{q}, \mathbf{a}) = \frac{\sum_{k=1}^{n} w_{q_k} \times w_{a_k}}{\sqrt{\sum_{k=1}^{n} w_{q_k}^2} \times \sqrt{\sum_{k=1}^{n} w_{a_k}^2}} \quad (4)$$

where w_{q_k} and w_{a_k} stand for the weight of the kth word in the question and the answer respectively.

[5]Code is available at http://www.kyb.tuebingen.mpg.de/bs/people/carl/code/minimize/

The weights can be get by computing the product of term frequency (*tf*) and inverse document frequency (*idf*)

HowNet based Similarity. HowNet[6] is an electronic world knowledge system, which serves as a powerful tool for meaning computation in human language technology. Normally the similarity between two passages can be calculated by two steps: (1) matching the most semantic-similar words in each passages greedily using the API's provided by HowNet; (2) computing the weighted average similarities of the word pairs. This strategy is taken as a baseline method for computing the relevance between questions and answers.

KL-divergence Language Model. Given a question **q** and its candidate answer **a**, we can construct unigram language model M_q and unigram language model M_a. Then we compute KL-divergence between M_q and M_a as below:

$$KL(M_a||M_q) = \sum_w p(w|M_a) \log(p(w|M_a)/p(w|M_q)) \quad (5)$$

5.2 Results and Analysis

We evaluate the performance of our approach for answer detection using two metrics: Precision@1 (P@1) and Mean Reciprocal Rank (MRR). Applying the two metrics, we perform the baseline methods and our DBN based methods on the two testing set above.

Table 1 lists the results achieved on the forum data using the baseline methods and ours. The additional "Nearest Answer" stands for the method without any ranking strategies, which returns the nearest candidate answer from the question by position. To illustrate the effect of the fine-tuning for our model, we list the results of our method without fine-tuning and the results with fine-tuning.

As shown in Table 1, our deep belief network based methods outperform the baseline methods as expected. The main reason for the improvements is that the DBN based approach is able to learn semantic relationship between the words in QA pairs from the training set. Although the training set we offer to the network comes from a different source (the cQA corpus), it still provide enough knowledge to the network to perform better than the baseline methods. This phenomena indicates that the homogenous corpora for training is effective and meaningful.

Method	P@1 (%)	MRR (%)
Nearest Answer	21.25	38.72
Cosine Similarity	23.15	43.50
HowNet	22.55	41.63
KL divergence	25.30	51.40
DBN (without FT)	**41.45**	**59.64**
DBN (with FT)	**45.00**	**62.03**

Table 1: Results on Forum Dataset

We have also investigated the reasons for the unsatisfying performance of the baseline approaches. Basically, the low precision is ascribable to the forum corpus we have obtained. As mentioned in Section 1, the contents of the forum posts are short, which leads to the sparsity of the features. Besides, when users post messages in the online forums, they are accustomed to be casual and use some synonymous words interchangeably in the posts, which is believed to be a significant situation in Chinese forums especially. Because the features for QA pairs are quite sparse and the content words in the questions are usually morphologically different from the ones with the same meaning in the answers, the Cosine Similarity method become less powerful. For HowNet based approaches, there are a large number of words not included by HowNet, thus it fails to compute the similarity between questions and answers. KL-divergence suffers from the same problems with the Cosine Similarity method. Compared with the Cosine Similarity method, this approach has achieved the improvement of 9.3% in P@1, but it performs much better than the other baseline methods in MRR.

The baseline results indicate that the online forum is a complex environment with large amount of noise for answer detection. Traditional IR methods using pure textual features can hardly achieve good results. The similar baseline results for forum answer ranking are also achieved by Hong and Davison (2009), which takes some non-textual features to improve the algorithm's performance. We also notice that, however, the baseline methods have obtained better results on forum corpus (Cong et al., 2008). One possible reason is that the baseline approaches are suitable for their data, since we observe that the "nearest answer" strategy has obtained a 73.5% precision in their work.

Our model has achieved the precision of

[6]Detail information can be found in: http://www.keenage.com/

45.00% in P@1 and 62.03% in MRR for answer detecting on forum data after fine-tuning, while some related works have reported the results with the precision over 90% (Cong et al., 2008; Hong and Davison, 2009). There are mainly two reasons for this phenomena: Firstly, both of the previous works have adopt non-textual features based on the forum structure, such as *authorship*, *position* and *quotes*, etc. The non-textual (or social based) features have played a significant role in improving the algorithms' performance. Secondly, the quality of corpora influences the results of the ranking strategies significantly, and even the same algorithm may perform differently when the dataset is changed (Hong and Davison, 2009). For the experiments of this paper, large amount of noise is involved in the forum corpus and we have done nothing extra to filter it.

Table 2 shows the experimental results on the cQA dataset. In this experiment, each sample is composed of one question and its following several candidate answers. We delete the ones with only one answer to confirm there are at least two candidate answers for each question. The candidate answers are rearranged by post time, so that the real answers do not always appear next to the questions. In this group of experiment, no hand-annotating work is needed because the real answers have been labeled by cQA users.

Method	**P@1 (%)**	**MRR (%)**
Nearest Answer	36.05	56.33
Cosine Similarity	44.05	62.84
HowNet	41.10	58.75
KL divergence	43.75	63.10
DBN (without FT)	**56.20**	**70.56**
DBN (with FT)	**58.15**	**72.74**

Table 2: Results on cQA Dataset

From Table 2 we observe that all the approaches perform much better on this dataset. We attribute the improvements to the high quality QA corpus Baidu Zhidao offers: the candidate answers tend to be more formal than the ones in the forums, with less noise information included. In addition, the "Nearest Answer" strategy has reached 36.05% in P@1 on this dataset, which indicates quite a number of askers receive the real answers at the first answer post. This result has supported the idea of introducing position features. What's more, if the best answer appear immediately, the asker tends to lock down the question thread, which helps to reduce the noise information in the cQA corpus.

Despite the baseline methods' performances have been improved, our approaches still outperform them, with a 32.0% improvement in P@1 and a 15.3% improvement in MRR at least. On the cQA dataset, our model shows better performance than the previous experiment, which is expected because the training set and the testing set come from the same corpus, and the DBN model is more adaptive to the cQA data.

We have observed that, from both of the two groups of experiments, fine-tuning is effective for enhancing the performance of our model. On the forum data, the results have been improved by 8.6% in P@1 and 4.0% in MRR, and the improvements are 3.5% and 3.1% individually.

6 Conclusions

In this paper, we have proposed a deep belief network based approach to model the semantic relevance for the question answering pairs in social community corpora.

The contributions of this paper can be summarized as follows: (1) The deep belief network we present shows good performance on modeling the QA pairs' semantic relevance using only word features. As a data driven approach, our model learns semantic knowledge from large amount of QA pairs to represent the semantic relevance between questions and their answers. (2) We have studied the textual similarity between the cQA and the forum datasets for QA pair extraction, and introduce a novel learning strategy to make our method show good performance on both cQA and forum datasets. The experimental results show that our method outperforms the traditional approaches on both the cQA and the forum corpora.

Our future work will be carried out along two directions. Firstly, we will further improve the performance of our method by adopting the non-textual features. Secondly, more research will be taken to put forward other architectures of the deep networks for QA detection.

Acknowledgments

The authors are grateful to the anonymous reviewers for their constructive comments. Special thanks to Deyuan Zhang, Bin Liu, Beidong Liu and Ke Sun for insightful suggestions. This work is supported by NSFC (60973076).

References

Adam Berger, Rich Caruana, David Cohn, Dayne Freitag, and Vibhu Mittal. 2000. Bridging the lexical chasm: Statistical approaches to answer-finding. In *In Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 192–199.

Delphine Bernhard and Iryna Gurevych. 2009. Combining lexical semantic resources with question & answer archives for translation-based answer finding. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 728–736, Suntec, Singapore, August. Association for Computational Linguistics.

Gao Cong, Long Wang, Chin-Yew Lin, Young-In Song, and Yueheng Sun. 2008. Finding question-answer pairs from online forums. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 467–474, New York, NY, USA. ACM.

Shilin Ding, Gao Cong, Chin-Yew Lin, and Xiaoyan Zhu. 2008. Using conditional random fields to extract contexts and answers of questions from online forums. In *Proceedings of ACL-08: HLT*, pages 710–718, Columbus, Ohio, June. Association for Computational Linguistics.

Huizhong Duan, Yunbo Cao, Chin-Yew Lin, and Yong Yu. 2008. Searching questions by identifying question topic and question focus. In *Proceedings of ACL-08: HLT*, pages 156–164, Columbus, Ohio, June. Association for Computational Linguistics.

Donghui Feng, Erin Shaw, Jihie Kim, and Eduard H. Hovy. 2006. An intelligent discussion-bot for answering student queries in threaded discussions. In Ccile Paris and Candace L. Sidner, editors, *IUI*, pages 171–177. ACM.

G. E. Hinton and R. R. Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

Georey E. Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14.

Liangjie Hong and Brian D. Davison. 2009. A classification-based approach to question answering in discussion boards. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 171–178, New York, NY, USA. ACM.

Jizhou Huang, Ming Zhou, and Dan Yang. 2007. Extracting chatbot knowledge from online discussion forums. In *IJCAI'07: Proceedings of the 20th international joint conference on Artifical intelligence*, pages 423–428, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Jiwoon Jeon, W. Bruce Croft, and Joon Ho Lee. 2005. Finding similar questions in large question and answer archives. In *CIKM '05*, pages 84–90, New York, NY, USA. ACM.

Jiwoon Jeon, W. Bruce Croft, Joon Ho Lee, and Soyeon Park. 2006. A framework to predict the quality of answers with non-textual features. In *SIGIR '06*, pages 228–235, New York, NY, USA. ACM.

Valentin Jijkoun and Maarten de Rijke. 2005. Retrieving answers from frequently asked questions pages on the web. In *CIKM '05*, pages 76–83, New York, NY, USA. ACM.

Jung-Tae Lee, Sang-Bum Kim, Young-In Song, and Hae-Chang Rim. 2008. Bridging lexical gaps between queries and questions on large online q&a collections with compact translation models. In *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 410–418, Morristown, NJ, USA. Association for Computational Linguistics.

Fangtao Li, Yang Tang, Minlie Huang, and Xiaoyan Zhu. 2009. Answering opinion questions with random walks on graphs. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 737–745, Suntec, Singapore, August. Association for Computational Linguistics.

Stefan Riezler, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu Mittal, and Yi Liu. 2007. Statistical machine translation for query expansion in answer retrieval. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 464–471, Prague, Czech Republic, June. Association for Computational Linguistics.

Ruslan Salakhutdinov and Geoffrey Hinton. 2009. Semantic hashing. *Int. J. Approx. Reasoning*, 50(7):969–978.

Lokesh Shrestha and Kathleen McKeown. 2004. Detection of question-answer pairs in email conversations. In *Proceedings of Coling 2004*, pages 889–895, Geneva, Switzerland, Aug 23–Aug 27. COLING.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2008. Learning to rank answers on large online QA collections. In *Proceedings of ACL-08: HLT*, pages 719–727, Columbus, Ohio, June. Association for Computational Linguistics.

Baoxun Wang, Bingquan Liu, Chengjie Sun, Xiaolong Wang, and Lin Sun. 2009. Extracting chinese question-answer pairs from online forums. In *SMC 2009: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2009.*, pages 1159–1164.