Mirror on the Wall: Finding Similar Questions with Deep Structured Topic Modeling

by

arpita.das , Manish Shrivastava, Manoj Chinnakotla

in

The 20th Pacific Asia Conference on Knowledge Discovery and Data Mining (PAKDD) 2016

The University of Auckland Auckland, New Zealand

Report No: IIIT/TR/2016/-1

Mirror on the Wall: Finding Similar Questions with Deep Structured Topic Modeling

Arpita Das[1], Manish Shrivastava[1], and Manoj Chinnakotla[2]

[1] International Institute of Information Technology Hyderabad, India
[2] Microsoft, India
arpita.das@research.iiit.ac.in,
m.shrivastava@iiit.ac.in,
manojc@microsoft.com

Abstract. Internet users today prefer getting precise answers to their questions rather than sifting through a bunch of relevant documents provided by search engines. This has led to the huge popularity of Community Question Answering (cQA) services like Yahoo! Answers, Baidu Zhidao, Quora, StackOverflow *etc.*, where forum users respond to questions with precise answers. Over time, such cQA archives become rich repositories of knowledge encoded in the form of questions and user generated answers. In cQA archives, retrieval of similar questions, which have already been answered in some form, is important for improving the effectiveness of such forums. The main challenge while retrieving similar questions is the "lexico-syntactic" gap between the user query and the questions already present in the forum. In this paper, we propose a novel approach called "Deep Structured Topic Model (DSTM)" to bridge the lexico-syntactic gap between the question posed by the user and forum questions. DSTM employs a two-step process consisting of initially retrieving similar questions that lie in the vicinity of the query and latent topic vector space and then re-ranking them using a deep layered semantic model. Experiments on large scale real-life cQA dataset show that our approach outperforms the state-of-the-art translation and topic based baseline approaches.

Keywords: Community Question Answering, Machine Learning, Deep Neural Networks

1 Introduction

Community Question Answering (cQA) services such as Yahoo! Answers, Baidu Zhidao, Quora, StackOverflow *etc.* build a community of "experts" on various topics of interest. They provide a platform where any internet user can pose a question on specific topics and get answers from other users or experts. Compared to search engines, cQA forums offer the following advantages - a) they provide the exact answer to user questions in stead of having to sift through a list of web pages b) they provide answers to opinion based questions which are usually answered by other users based on their personal experience. Though,

the answers provided by cQA users are usually of good quality, sometimes they can be noisy and redundant too. However, the noise in the data is offset by other forum meta-data associated with each question such as user provided ratings, categories, best answers ratings *etc.* Over time, cQA forums accumulate an archive of rich knowledge encoded in the form of user question and answers on various topics of interest.

The effectiveness of a cQA forum, depends on its ability to retrieve similar questions, which have already been answered, in response to new questions posed by users. This helps in - a) Saving the time and attention of experts/users by allowing them to focus on new unanswered questions in stead of repeated questions expressed in a different language. b) Minimizing the time lag of the user waiting for an answer c) Preventing question starvation where if a question goes unanswered for a long time, answers to similar questions could be offered to the user and d) Generating automated answers from the answers provided to similar questions in various cQA forums. The major challenge associated with this problem is bridging the lexico-syntactic gap. Two questions can convey the same meaning even though they are syntactically and lexically different. For example: "Can caffeine contents be harmful to health?" and "Does coffee have adverse effects on health?" are semantically similar.

In this paper, we address the above problem of retrieval of similar questions from cQA archives. Previous approaches in literature have relied on - a) learning word or phrase level translation models [4,7,9] from question-answer pairs in parallel corpora of same language. The similarity function between questions is then defined as the probability of translating a given question into another or b) learning topic models from question-answer pairs [5,2,8]. Here, the similarity between questions, is defined in the latent topic space discovered by the topic model. However, they suffer from the following short-comings - a) Large parallel sentence-aligned question corpora is usually a scarce resource. This limits the effectiveness of translation-based approaches. b) Topic based approaches usually perform well in improving recall but sacrifice on precision. Recently, Deep Structured Semantic Models (DSSM) [3] have shown state-of-the-art performance when compared to translation models and topic models for learning latent semantic representations. DSSM leverages large volumes of click-through data available to commercial search engines to learn a low-dimensional representation of the queries and documents. This is done by discriminatively training a neural network which maximizes the conditional likelihood of the clicked documents given a query. Further, Convolutional DSSM (CDSSM) adds a convolutional layer to the DSSM structure that projects each word within a context window to a local contextual feature vector. Semantically similar words within a context are projected to vectors that are close to each other in the contextual feature space.

In this paper, we propose a novel approach called "Deep Structured Topic Model (DSTM)" which leverages the strengths of both deep learning based methods and topic models. In the training phase, DSTM learns - a) A topic model, Latent Dirichlet Allocation (LDA) [1], for discovering the latent topics from

the question-answer archive and b) CDSSM semantic model for modeling the hidden semantic similarities between question-answer pairs. Given a new user question, DSTM employs a two-step process consisting of initially retrieving similar questions that lie in the vicinity of the query in the latent topic vector space discovered by LDA and then re-ranking them using the CDSSM model. Since, topic models such as LDA are expected to have better recall, the initial step will ensure that most relevant questions are filtered from the entire corpus. Later, this filtered set is re-ranked using CDSSM such that the most semantically similar questions are ranked higher in the order. Thus, we combine the strengths of both topic models and deep semantic models. Our contributions are (1) We enhance the topic modeling approach using deep neural networks. We employ deep semantic models to capture semantic similarity between the question pairs with same topic distribution. (2) Qualitative and quantitative analysis of our results using large scale Yahoo! dataset and comparison with other state-of-the-art methods.

The rest of the paper is organized as follows. Section 2 presents our work in the context of related work. Section 3 describes the process of filtering relevant questions from corpus using LDA. Section 4 gives more details on Deep Semantic Modeling. Section 5 gives details of our system such as architecture and training methodology. Section 6 describes the experimental set-up, details of the evaluation datasets, evaluation metrics and also presents the quantitative and qualitative results. Finally, Section 7 concludes the paper.

2   Related Work

[4] visualized question retrieval problem as word-based translation problem. They utilized the similarity between answers in the archives to estimate translation probabilities. [7] incorporated query likelihood language model and word based translation model to enhance the performance. In order to capture contextual information, [9] proposed phrase based translation model for similar question retrieval. These models use question answer pairs as parallel corpus.

Another way of approaching this retrieval problem is topic modeling. [2] exploited latent topic information of the question content. They infused "category"(every question in cQA data have a category and a set of sub-categories) information in the topic space to find the questions semantically nearer to the query. [5] proposed Question-Answer Topic Model (QATM) to learn the latent topics aligned across the question-answer. They also introduced QATM with posterior regularization (QATM-PR) to prevent the topic vector representation of question-answer pair to be dominated by the answers which are much longer in length than the questions. [8] used supervised question-answer topic model. They infused language model with topic model for matching the questions on both term and topic levels.

[10] learned word embeddings of questions with "category" information. They employed a fisher kernel to convert the variable length question vectors to a fixed
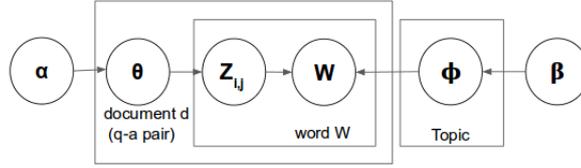
Fig. 1. Graphical model of generative LDA.

size representation. Our work is aligned towards the topic modeling and deep semantic models [3,6].

3   LDA based selection of Relevant Questions

Traditionally, LDA [1] has given good results for related question selection in cQA forums [2,5,8]. LDA assumes that a document exhibits multiple topics. In this problem setting each document is a question and its best answer pair. Assume, there are M documents in the corpus and every document is composed of words W. N is the total number of words in a particular document and K is the total number of topics to be considered. LDA is a probabilistic model where observed variables are the bag of words per document and the hidden random variables are θ,ϕ and $Z(i,j)$ where θ is the topic distribution per document, ϕ is the distribution of the vocabulary per topic and $Z(i,j)$ is the topic of the jth word in the ith document . It can be considered as a unsupervised model where documents are unlabeled and no information of topic distribution is known beforehand. Our goal is to calculate the posterior probability of the hidden variables given the observations. The graphical model of LDA can be seen in Figure 1. Documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. The generative process of LDA is used to find the joint probability of the observed variables and hidden variables. It is carried out as follows:

1. Choose $\theta_i \sim \text{Dir}(\alpha)$ where $i \in \{ 1,\ldots,M \}$ and $\text{Dir}(\alpha)$ is the Dirichlet distribution for parameter α
2. Choose $\phi_k \sim \text{Dir}(\beta)$ where $k \in \{ 1,\ldots,K \}$
3. For each word position (i, j) where $j \in \{ 1,\ldots,N_i \}$, and $i \in \{ 1,\ldots,M \}$
    (a) Choose a topic $z_{i,j} \sim \text{Multinomial}(\theta_i)$.
    (b) Choose a word $w_{i,j} \sim \text{Multinomial}(\phi_{z_{i,j}})$.

After finding the joint probability of hidden and observed variables we can find the posterior or conditional probability of the hidden variable given the observed one by equation 1.

$$P(\beta_{1:K}, \theta_{1:K}, Z_{1:K}|w_{1:M}) = \frac{P(\beta_{1:K}, \theta_{1:K}, Z_{1:K}, w_{1:M})}{P(w_{1:M})} \qquad (1)$$

The posterior probability is used to obtain the topic vector distribution of each document. Thus, each document can be represented in latent topic space. But, considering the noisy and redundant nature of data found in cQA sites it would not be a good idea to assume that semantically identical questions will have similar topic distribution.

As shown in Table 1, few questions retrieved for the query "How the human species evolved?" like "Who is the most mature species in planet?" or "How old is human being?" are not semantically identical to the query in spite of having similar topic distribution. This means that topic models may not be capturing the relevant semantic content of the query entirely. To rectify this problem we need to employ a model which takes deeper semantic similarity into account. This leads us to develop DSTM.

4   Deep Semantic Modeling

DSSM [3] and CDSSM [6] are the deep neural networks that map raw textual features into vectors in semantic space. [3] have shown that these models work with click-through data consisting of query and set of clicked documents. It is interesting to note that the input to these networks is not high dimensional term vectors of documents. Instead, word hashing involving letter n-gram is used to reduce the dimensionality of term vectors. For a word, say, "book" represented as (#book#) where # is the delimiter used, letter 3-grams would be #bo, boo, ook, ok#. Since the size of training data used is in millions, representing every word with one hot vector would be practically infeasible. Word hashing allows us to represent a query or a document using a lower dimensional vector (dimension equal to number of unique letter trigrams obtained in the corpus). It also takes care of out-of-vocabulary words and words with spelling errors.

DSSM uses multiple hidden layers to project the word-hashed features into the semantic space. The semantic relatedness between query and document is scored on the basis of cosine similarity between them. DSSM proposes a supervised training method to learn the weight matrices and the bias vectors of the deep neural network (DNN). The objective of DSSM is to maximize the conditional likelihood of the clicked documents given the queries or to minimize the loss function in equation 2.

$$L(\Lambda) = -log \prod_{(Q,D^+)} P(D^+|Q) \tag{2}$$

where Λ denotes the set of parameters of the DNN, Q is the query, D^+ is the set of documents clicked for query Q and $P(D^+|Q)$ is the posterior probability of a clicked document given a query from the semantic relevance score between them using a softmax function.

CDSSM [6] introduces a convolutional layer in addition to DSSM. The advantage of CDSSM is that it considers words at a contextual level and projects each word within a context to a local contextual feature vector. Further, CDSSM
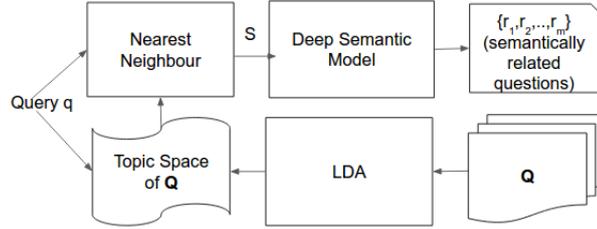
Fig. 2. System architecture of DSTM model.

uses a max pooling layer to extract the crucial local features to form a fixed-length global feature vector. Affine transformations and element-wise non-linear functions is then applied to the global feature vectors to extract highly non-linear and effective features. Both DSSM and CDSSM captures the non-linear semantic structures between a query and a document. These models significantly outperformed the existing Web retrieval models.

5   Deep Structured Topic Modeling (DSTM)

Inspired by the success of DSSM and CDSSM in the fields of web retrieval, we propose DSTM or Deep Structured Topic Model which aims to capture non linear semantic relatedness and topic information between questions in cQA forums. The system architecture is showed in Figure 2.

We use LDA to calculate the latent topic distribution of each document (a question and its best answer) in our dataset. Each question Q is represented with a topic vector of the question-best answer pair (Q-BA) obtained using LDA. The question Q in the document d has a topic space representation: [$t_1, t_2, \ldots, t_K$] where $t_j \in [0,1]$ is the indicator variable denoting whether the topic t_j is assigned to document or not, $j = \{1,2,\ldots,K\}$ and K is the total number of topics. Given a query q we find its n nearest neighbors in the topic space using cosine similarity. The candidate set is $C = \{r_1, r_2, \ldots, r_n\}$, where each r_i is a relevant candidate question retrieved. Set S is the set of pairs of queried question and the retrieved candidates r_i present in C.

$$S = \{(q, r_1), (q, r_2), \ldots, (q, r_n)\} \tag{3}$$

As explained earlier, questions with same topic distribution may not be semantically similar thus we employ DNN to capture non linear semantic relations between them.

The term vector of question pairs in S are word hashed [3] and fed to the DNN. It maps the word hashed vectors to their corresponding semantic concept vectors. For testing, the term vector of retrieved question r_i is calculated using both the question and its answer (best answer and most voted answers) pair. The similarity between the query q and a candidate question r_i is calculated using cosine similarity of their corresponding semantic concept vectors.
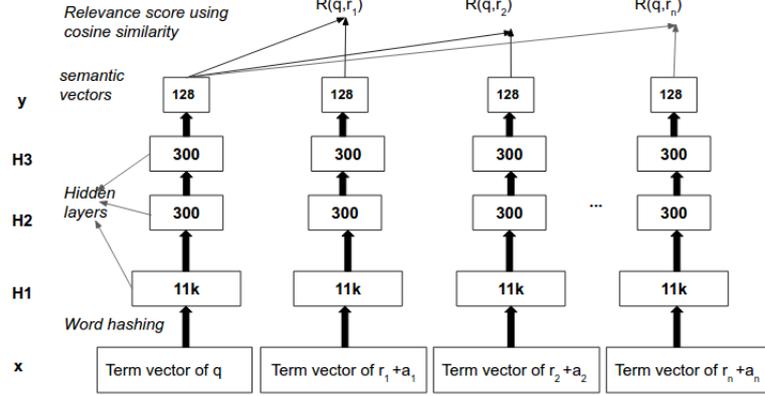
Fig. 3. Deep semantic model of DSTM. Here q is the new query and $r_i + a_i$ are the nearest question-answer pairs given by LDA.

Let x be the word hashed input term vector, y is the output vector and h is the number of hidden layers used. Let, H_j represents the jth intermediate layer whose weight matrix is W_j and bias term is b_j, where $j = \{1, 2,\ldots,h\}$.

$$l_j = f(W_j H_{j-1} + b_j) \tag{4}$$

where $j = \{2,3,...,h\}$ and $H_1 = W_1 x$

$$y = f(W_h H_{h-1} + b_h) \tag{5}$$

where we use tanh as the activation function.

$$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{6}$$

The relevance score between query and candidate questions is :

$$R(q, r_i) = \frac{y_q^T y_{r_i}}{|y_q||y_{r_i}|} \tag{7}$$

The training of neural network is explained in the next section. After the training is complete the questions pairs in S are fed to the network. We select the questions with the highest relevance scores with the query. For a query we output those questions that are semantically similar in the topic space. The testing phase of deep neural network in DSTM is shown in Figure 3.

For Convolutional Deep Structured Topic Model (CDSTM) we use a convolutional layer that captures local context of word within a sliding window. Every word of a document is represented by a count vector of its letter-trigrams. The word hashed vectors of the words in a window are concatenated to form a context window vector. Global feature vector of each query or document is obtained by max pooling which retains only the important features from each context window. Non linear projections of global feature gives the semantic vector. The remaining operations are the same as in DSTM.

5.1   Training the Network

Large-scale standard dataset of similar questions is not available. Therefore, we cannot train the network to maximize the conditional likelihood of a similar question given the question. However, richer information is available to us in cQA data in terms of questions, associated answers and meta-data associated with them. We train the network for a question while looking for semantic similarity with their best answers. We train our deep neural network with the questions and their answer pairs (we consider only best answer and most voted answers as valid answers to the question). We learn the parameters of our network W_i and b_i to maximize the conditional likelihood of the corresponding answers given the question. The posterior probability of the answer given the question is:

$$P(A|Q) = \frac{\exp\left(\gamma R(Q, A)\right)}{\sum_{A' \in A} \exp\left(\gamma R(Q, A')\right)} \qquad (8)$$

where γ is the smoothing factor of the softmax function, **A** is the set of user-given answers to question Q.

Ideally all the answers provided to the query must be considered, but since the answers are human generated there is a lot of noise involved with them. For example, a query "Whats the best way to delete a computer virus?" (in Yahoo! Answers) has answers like "setup your computer.", "throw it out your window.", "Throw the computer in the garbage and buy a new one." etc. These answers are irrelevant and do not give any valid solution to the problem. So, in order to filter out noise and redundancy, we approximate **A** using A^+ which contains the best answer and the most voted answers for a query Q and A^- which contains any five answers from a random question other than Q. Finally, the problem boils down to the minimization of the loss function in equation 9.

$$L(\Lambda) = -log \prod_{(Q, A^+)} P(A^+|Q) \qquad (9)$$

6   Experiments

We collected Yahoo! Answers dataset from Yahoo! Labs Webscope. It has about 5 million questions where each question contains title, description, best answer, most voted answers and meta-data like categories , sub categories etc. The questions in this dataset are distributed across a number of categories. Each question was pre-processed by lower-casing, stemming, stopword and special character removal.

We randomly selected a set of thousand questions from the above Yahoo! dataset as our test set. In order to obtain the true labels for the test set, we split the test set into 10 batches containing 100 questions each. For each question in a batch, we retrieved the top 20 similar questions as options using BM25 ranking algorithm in Lucene scoring function. Each batch of questions was distributed to three different human judges. Each judge was asked to vote for the most relevant

question expressing the same intent from the provided options. Since each batch was distributed to three different annotators, majority of their votes was taken to judge relevance of a question given a query.

On this gold data, we evaluate the performance of all the models using four evaluation criteria: Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), R-Precision (R-Prec) and Precision at K (P@K).

6.1   Results

In this section, a comparative study of the previous methods is done with respect to our method. The baseline methods considered are query likelihood language model (LM) and BM25. To show how translation based methods perform we implemented the papers [4,7,9]. [4] deals with word based translation, [7] enhanced the first by adding language model to it and [9] implements phrase based translation method. As seen in Table 2, the translation based methods outperform the baseline significantly. The models are trained using GIZA++ tool with the question and best answer pair as the parallel corpus. We also implemented the papers using unsupervised and supervised topic modeling approaches presented in [5,8] respectively. The topic-based models form the current state-of-the-art. Again it is visible from the Table 2 topic based approaches shows comparable performance with translation based methods but they show significant improvement over baseline. We have used Sent2Vec to evaluate the results shown by DSSM and CDSSM [3] and [6]. These models outperforms the traditional methods but DSTM and CDSTM outperforms them. This proves that latent topic information enhances the deep semantic models in [3] and [6].

We also evaluated DSTM and CDSTM models in the dataset released by [8]. The comparison of performance are shown in Table 3. The better performance of DSTM and CDSTM on this new dataset proves their stability. Our method which combines topic modeling and deep semantic pruning outperforms the other approaches significantly. The improvement could be attributed to two factors - 1) We assume that semantically similar questions may not have similar topic distribution and 2) We model the non-linear semantic dependencies between the question and best answer pair and utilize it to output the most relevant similar questions given a query.

6.2   Qualitative Analysis

In Table 1, examples are shown to depict performance of supervised topic based method and DSTM. In Q1, the questions that DSTM outputs are semantically much closer, for example "How our society has evolved?" or "Does history of human evolution begin from two persons?". These are very relevant to the person who wants to know "How the human species evolved?". In Q2, the query is about the best way to boot a Windows CD. The questions retrieved by both the models are related to "computers", "booting" etc but DSTM really understands that the question is about making a "Windows" CD bootable. DSTM is able to capture that Win98se, WinXP, WinProfessional are related to Windows. In Q3, the user

Q1: How the human species evolved?	
Supervised Topic Model	1.Except human, what is the most evolved and intelligent species on this planet? 2.How old is the human species ? 3.Who is the most mature species on the planet?
Our Model	1.How our society has evolved? 2.Does history of human evolution begin from two persons? 3.Is it real that humans evolved from monkeys ?
Q2: What's the best way to create a bootable windows CD?	
Supervised Topic Model	1.How do I create a bootable winxp dvd? 2.How can I make a cd from the C-drive? 3.How to make a bootable OS CD?
Our Model	1.How to make windows bootable cd for win98se? 2.How do I create a bootable winxp dvd? 3.How can I create a bootable winxppro cddisk?
Q3: I am looking for a new bike... what's the best bike for on-road exercise?	
Supervised Topic Model	1.What should be my new bike? 2.What is the maintenance for my new bike? 3.What is the best website to buy a new bike?
Our Model	1.What should I buy for a mountain bike to get into sports? 2.Buying a bike, any ideas? 3.Looking for a two seater?
Q4: What's the least expensive way to make a movie using local talent ?	
Supervised Topic Model	1.What was the most expensive movie to make ? 2.How can I make a movie with a camcorder and a bunch of friends ? 3.Hi, I need to make a movie of my times in college. Please help..
Our Model	1.What was the most expensive movie to make ? 2.What is the basic thing to make a movie ? 3.Will they ever make a dark tower movie prolly someday ?

Table 1. This table compares the performance of supervised topic model and DSTM. DSTM performs better in Q1-3 while topic model shows better performance in Q4.

want suggestions for buying a new bike, the supervised topic model retrieves question that deals with a bike in general i.e., queries dealing with maintenance and website of new bikes are retrieved. DSTM model output questions that are exclusive about buying a new bike, also it is interesting to note that it can capture that "two seater" and bike are semantically similar.

Q4 is an instance where the performance of supervised topic model is better than DSTM. Both the models understand that the query is about making movies but most of the questions retrieved by topic model are more aligned towards how movies can be made in the least expensive way (using camrecorder, casting friends, making college life movie etc) than DSTM.

Method	MAP	MRR	R-Prec	P@5
LM	0.355	0.392	0.238	0.204
BM25	0.360	0.399	0.275	0.262
translation(word) [4]	0.385	0.431	0.301	0.292
translation+LM [7]	0.399	0.464	0.296	0.343
translation(phrase) [9]	0.479	0.486	0.380	0.361
Q-A topic model [5]	0.463	0.477	0.365	0.351
Q-A topic model(s) [8]	0.474	0.482	0.368	0.359
DSSM[3]	0.513	0.659	0.388	0.386
CDSSM[6]	0.515	0.661	0.387	0.392
DSTM	0.531	0.574	0.392	0.401
CDSTM	**0.532**	**0.574**	**0.393**	**0.420**

Table 2. Results on Yahoo dataset. The best results obtained by CDSTM are bold faced. The difference between the results of DSTM,CDSTM and other methods are statistically significant with $p < 0.05$.

Method	MAP	MRR	R-Prec	P@5
$TBLM_{SQATM}$[8]	0.805	0.889	0.718	0.831
DSTM	0.834	0.903	0.759	0.870
CDSTM	**0.836**	**0.912**	**0.759**	**0.877**

Table 3. Results on dataset released by [8]. $TBLM_{SQATM}$ is the model introduced by [8]. The best results obtained by CDSTM are bold faced.

7   Conclusion

In this paper, we proposed a novel approach called "Deep Structured Topic Model (DSTM)" to bridge the lexico-syntactic gap between the question posed by the user and forum questions. DSTM employs a two-step process consisting of initially retrieving similar questions that lie in the vicinity of the query and latent topic vector space and then re-ranking them using a deep layered semantic model. We showed that questions with same topic distribution may not be semantically identical and hence deep semantic models help in enhancing the performance of topic models. We evaluated our approach using large scale datasets from real-world cQA forums and showed that DSTM and CDSTM performs better than other state-of-the-art baseline approaches. Qualitative analysis of our results reveals that DSTM is indeed able to leverage the advantages of both topic modeling approaches such as LDA and also deep semantic models such as CDSSM. As part of future work, we would like to combine the meta-data information and the topic vector representation of the questions to train DSTM and CDSTM to find out if meta-data can enhance our existing models.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
2. Cai, L., Zhou, G., Liu, K., Zhao, J.: Learning the latent topics for question retrieval in community QA. In: Fifth International Joint Conference on Natural Language Processing, IJCNLP 2011. pp. 273–281 (2011)
3. Huang, P., He, X., Gao, J., Deng, L., Acero, A., Heck, L.P.: Learning deep structured semantic models for web search using clickthrough data. In: 22nd ACM International Conference on Information and Knowledge Management, CIKM'13. pp. 2333–2338 (2013)
4. Jeon, J., Croft, W.B., Lee, J.H.: Finding similar questions in large question and answer archives. In: Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management. pp. 84–90. ACM (2005)
5. Ji, Z., Xu, F., Wang, B., He, B.: Question-answer topic model for question retrieval in community question answering. In: 21st ACM International Conference on Information and Knowledge Management, CIKM'12. pp. 2471–2474 (2012)
6. Shen, Y., He, X., Gao, J., Deng, L., Mesnil, G.: Learning semantic representations using convolutional neural networks for web search. In: 23rd International World Wide Web Conference, WWW. pp. 373–374 (2014)
7. Xue, X., Jeon, J., Croft, W.B.: Retrieval models for question and answer archives. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008. pp. 475–482 (2008)
8. Zhang, K., Wu, W., Wu, H., Li, Z., Zhou, M.: Question retrieval with high quality answers in community question answering. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM. pp. 371–380 (2014)
9. Zhou, G., Cai, L., Zhao, J., Liu, K.: Phrase-based translation model for question retrieval in community question answer archives. In: The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference. pp. 653–662 (2011)
10. Zhou, G., He, T., Zhao, J., Hu, P.: Learning continuous word embedding with metadata for question retrieval in community question answering. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015. pp. 250–259 (2015)