Text-Based Question Routing for Question Answering Communities via Deep Learning

Amr Azzam and Neamat Tazi
Faculty of computers and Infomation
Cairo University
Cairo, Egypt
a.tarek, n.eltazi@Fci-cu.edu.eg

Ahmad Hossny
School of mathematical sciences,
University of Adelaide,
Adelaide, Australia,
ahmad.hossny@adelaide.edu.au

ABSTRACT

Online Communities for Question Answering (CQA) such as Quora and Stack Overflow face the challenge of providing sufficient answers for the questions asked by users. The exponential growing rate of the unanswered questions compromises the effectiveness of the CQA frameworks as knowledge sharing platforms. The main reason for this issue is the inefficient routing of the questions to the potential answerers, who are the field experts and interested users.

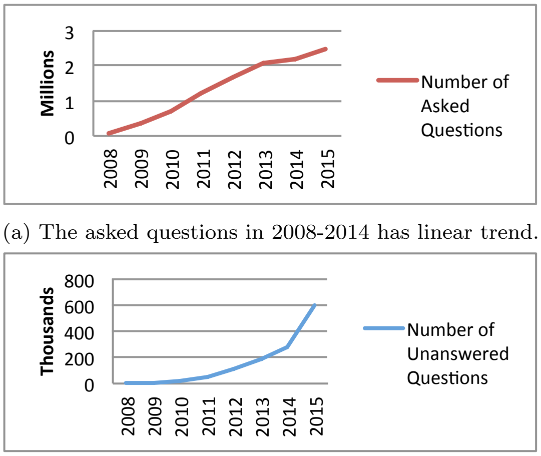
This paper proposes the deep-learning-based technique QR-DSSM to increase the accuracy of the question routing process. This technique uses deep semantic similarity model (DSSM) to extract semantic similarity features using deep neural networks and use the features to rank users' profiles . QR-DSSM maps the asked questions and the profiles of the users into a latent semantic space where the ability to answer is measured using the cosine similarity between the questions and the profiles of the users. QR-DSSM experiments outperformed the baseline models such as LDA, SVM, and Rank-SVM techniques and achieved an MRR score of 0.1737.

CCS Concepts

•**Information systems** → **Retrieval models and ranking;**

Keywords

Question Routing; Community Question Answering; Deep Learning; Semantic Modeling



(a) The asked questions in 2008-2014 has linear trend.

(b) The unanswered questions in 2008-2014 has an exponential trend.

Figure 1: Stack Overflow Website statistics

1 Introduction

Community Question Answering (CQA) websites are a major source for information seeking. CQA portals such as (Yahoo! Answers, Quora and Stack Exchange) give the users the advantage of having an answer for the questions that need domain specific experience or require an implicit knowledge that is hard to obtain through regular search engines. A question can be routed using textual features or statistical features. Textual features require natural language processing such as morphological[9], syntax, and semantic analysis along with machine learning techniques [10].

Although CQA is an effective way for knowledge sharing, it has some limitations that reduce the participation of users in the community. The fast growing number of posted questions on a daily basis puts an extra overhead on the answerers for finding the appropriate questions that match their expertise. On the other hand, from the asker's perspective, they might receive low-quality answers from non-experts. Moreover, they face an increase of average waiting time for their questions to be answered by qualified answerers. These limitations lead to a reduction in the knowledge shared by experienced users and hence compromise the objective of these communities.

It is essential to CQA frameworks to have a question routing system that is able to automatically route the newly posted questions to the potential qualified answerers [18][15][11][13]. The time lag between posting a question and providing a high-quality answer should be minimized to encourage askers and answerers to participate in and rely on the CQA framework.

Stack Overflow is an online question answering framework for computer programming topics. The best answer for the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC 2017, April 3-7, 2017, Marrakesh, Morocco
Copyright 2016 ACM 978-1-4503-4486-9/17/04...$15.00
DOI:http://dx.doi.org/10.1145/3019612.3019762

question is selected by either the asker or by opening the question for voting. Stack Overflow statistics show that the platform has a continuously growing number of questions linearly but it suffers from the exponential increase in the number of unanswered questions as illustrated in figure 1.

In this paper, we propose a question routing technique called (QR-DSSM) that uses the textual features to predict the semantic similarity between the questions and profiles of the potential answerers using deep neural networks (DNN). The proposed method is compared with the LDA [4],SVM classifiers [5] and Rank-SVM [7], proposed in[11]. Also, the effect of removing the content of the coding elements from Stack overflow questions on the learning process is examined. QR-DSSM outperformed the mentioned techniques using the MRR and MAP metrics significantly.

QR-DSSM consists of three main phases as illustrated in Figure 2:

1. Data preparation phase: a user profile is created for each answerer in the community using his previously answered questions.

2. Learning phase: Measure the semantic similarity between the answerer profile and the newly posted questions using the deep structured semantic model (DSSM) [1] .

3. Decision-making phase: the question routing technique routes the question to a ranked list of potential answerers. The ranking of users is based on DSSM that produces a score using the cosine similarity between the question and the answerers' profiles.

The rest of the paper is organized as follows. Section 2 provides an overview of the existing question routing approaches. The proposed technique is explained in Section 3. In Section 4, experiments and evaluations are conducted to compare our technique with the existing ones. Finally, Section 5 presents the conclusion and future research directions based on this work.

2 Related Work: Question Routing Techniques

Question routing has two main approaches to represent the association between a posted question and potential answerers. The first approach is called profile based approach [15][11] where the textual representation of the answerer profile is based on the set of previously answered questions in the CQA.

The second approach is called document based approach which is a two step process. The first step finds the similarity between the newly posted question and the previously answered questions in the community [11]. The second step ranks the set of answerers of the previously answered questions based on the relevance of their answers history.

In the following subsections, we categorize these techniques according to the information retrieval approach used.

2.1 Vector Space Model (TF-IDF)

The authors in [13] proposed a framework that could find the right experts to a specific question or a category of questions. The framework builds a hybrid approach to create users' profiles. The authors in [13] created users' profiles based on vector space model to process past questions answered by the users. The users' profiles are converted to term vectors using TF-IDF approach [17]. Non-textual features were extracted and combined with the term vectors. Features such asthe user reputation in the community, link analysis approaches such as HITS [12] and Page Rank [14] are used to represent the user authority in the community. Moreover, other features were added such as the number of votes and the time factor that represents the importance of answered questions decays by time.

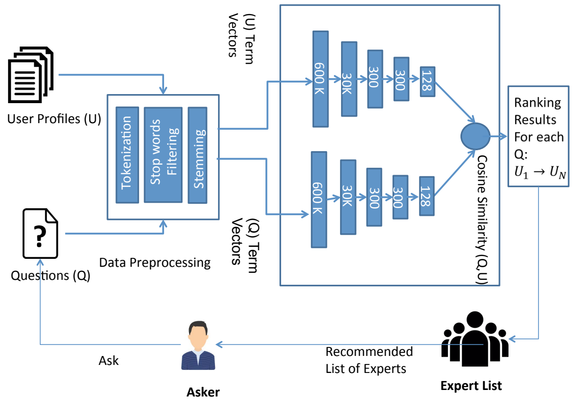


Figure 2: Proposed Question Routing Technique Stages

2.2 Topic Modeling

Topic modeling techniques discover hidden topics inside documents collections. Topic modeling proved its ability to find semantic similarity between documents even in lexically different ones. Many researchers applied these techniques to capture the similarity between posted questions and users' profiles [15]. The authors in [15] introduced a question routing framework based on a Segmented Topic Model (STM). The best-answered questions by each user were combined to create the users' profiles. As explained in [15], the authors compared STM with multiple information retrieval approaches such as TF-IDF, Language modeling, and LDA where STM outperformed the others.

2.3 Deep Learning

The authors in [16] proposed a two-stage semantic hashing approach for document retrieval. The first stage is a pre-training stage to learn the generative model, where the first layer is a word-count vector and the final layer is the document represented in binary code. The second stage is the fine-tuning stage where a deep auto-encoder network is constructed and back propagation is used to optimize the document reconstruction weights. Also, [2] proposed another semantic hashing model using deep belief nets. The first layer has word counts and the final layer has the document binary code and the remaining layers form a Bayesian network.

The authors in [6] introduced Bi-Lingual Topic Models (BLTM) and Linear Discriminative Projection Models (DPM). BLTM aimed to maximize the log-likelihood that is not effectively conformed with the ranked-based evaluation metrics. On the other hand, DPM has huge computation costs that make the training process with large vocabulary size infeasible.

3 Proposed Technque: Ranking User Profiles for Question Routing

The proposed model have utilized DSSM (deep structured semantic model) that was introduced in [1], to enhance users' profiles retrieval and ranking based on semantic features relevance to the posted questions rather than keyword/term relevance as [13] [17]. The main idea of DSSM is to use DNN to map the sparse raw text of the queries and the documents into dense features where the query and the documents can be represented in the same latent semantic space with a low dimension feature vectors. The documents are ranked based on maximizing the cosine similarity between extracted feature vectors.

In the following, a formal annotation is defined to be used in the network architecture explanation.

qx : Question input term vectors
ux : Users' profiles input term vectors
qxh: Question input vectors after word hashing
uxh: Users' profiles input vectors after word hashing
h: hidden layers
y: The output layer
Q: Question
U: User profile
W_i: The i^th weight matrix the bias term
b_i: The i^th bias term
n: The number of nodes in each layer of the network

The proposed technique takes the questions text Q and the user profile U as input and passes it through the cleansing stage, by removing web tags, code chunks and stopping words. The questions Q and users' profiles U are tokenized and converted to a sequence of words that generates two input term vectors qx and ux. This process results in sparse term vectors, which decreases the neural network efficiency and increases its computational cost. Afterwards, we apply word hashing embedding on qx and ux to reduce dimensionality[1]. Each word in the term vectors is decomposed into a vector of fragments where each fragment consists of three consecutive letters (tri-gram). For instance, the word "Hope" is represented as "#Hope#" where a delimiter is added in the beginning and the end of the word leading to the set of sub-words #Ho, Hop, ope, pe#. The output vectors of the word hashing are qxh and uxh where each vector contains the frequency of each tri-gram in the questions and users's profiles respectively.

The qxh and uxh vectors are passed to two networks with the same architecture. The architecture of the two DNN is a fully connected network topology. It consists of two hidden layers with n nodes each where n=300 nodes. The QR-DSSM can be described in mathematical formula as shown below:

$$y(k) = F\left(\sum_{i=0}^{m} W_i(k) \cdot xh_i(k) + b_i\right) \tag{1}$$

Where $xh_i(k)$ is the input value after hasing layer in discrete time k where i goes from 0 to m, $W_i(k)$ is the weight value in discrete time k, b is bias, F is the activation function, y is the output value in discrete time $W_i(k)$. The mathematical function used in this research as the activation function is $Tanh$ in both the output layers and the hidden layers.

$$F(x) = Tanh(x) = \frac{1 - e^{-2x}}{1 - e^{-2x}} \tag{2}$$

The output layer is constructed from n nodes where $n = 128$. The cosine similarity between the posted question Q and the users' profiles U is based on the output layer vectors of both networks y_Q and y_U

$$cosine(y_Q, y_Q) = \frac{y_Q^T \cdot y_U}{||y_Q|| \cdot ||y_U||} \tag{3}$$

The softmax function is implemented in the output layer of the network to convert the semantic relevance score between the posted question Q and the users' profiles U into a posterior probability of user profile given the question. It enforces an important constraint on the network outputs. The network output probability lies between 0 and 1 while its summation is 1.

$$\Pr(U \mid Q) = \frac{e^{cosine(y_Q, y_U)}}{\sum_{k=1}^{K} e^{cosine(y_Q, y_{U_k})}} \tag{4}$$

Log loss function is used to measure the accuracy of the users' profiles ranking given the posted questions. The model produces a probability for each user profile and the log loss function is the cross-entropy between the predicted outputs and the true user profile who actually provided an answer to the question.

The loss function is used to learn the weights by maximizing the product of the predicted probabilities. These weights represent a set of parameters (W_i, b_i) that the model should learn. The mathematical expression for the negative log prediction probability is as follows:

$$\Pr(U \mid Q) = L(W_i, b_i) = -log\left(\prod_{k=1}^{k} \Pr(U_k \mid Q)\right) \tag{5}$$

Where U_k is the set of users who actually provided an answer to the question

4 EXPERIMENTS AND RESULTS

4.1 Dataset

The dataset used in all experiments was constructed from a Stack Overflow data snapshot by following all conditions mentioned in [11]. Stack Overflow releases a dump data in XML format that contains the content of the website questions, answers, and users every quarter of a year for public use.

The data snapshot used in the following experiments spans 13 months starting from 1st of January, 2009 until 31st of January, 2010. We used the same subset of tags selected in [11][15] to select a set of questions that share the same characteristics. The selected questions should have an accepted answer and at least another answer.

The size of the dataset generated is 92,407 CQA sessions and it is divided into a training data and a test data based on the question creation date. The training data is the set of questions posted between 1st of January, 2009 and 31st of December, 2009 with size 81,295 questions. The test data is the set of the posted questions starting from 1st of January, 2009 and ends at 31st of January, 2010.

Table 1: User set, training set and test set numbers

# of questions answered by user U	# of Users U	# of Training Questions Q_{Trn}	# of Test Questions Q_{Tst}
10	5,759	16,021	1,151
15	3,970	11,177	746
20	2,977	8,371	517

Table 2: the number of pairs generated from pairing each question in Q_{Trn} with the answerers from the users (U).

N (number of questions answered by a user U)	# of Pairs in the training set
N = 10	54,218
N = 15	36,238
N = 20	26,354

4.2 Dataset Pre-processing

Several pre-processing actions were performed to clean the extracted dataset. These actions include HTML tags removal, tokenization and stop words filtration.

The output of these pre-processing actions is clean word vectors that are stemmed using stemming algorithms written by the Snowball language (https://github.com/snowballstem). Few questions in Stack Overflow contain blocks of code that are enclosed between <code>XML element.

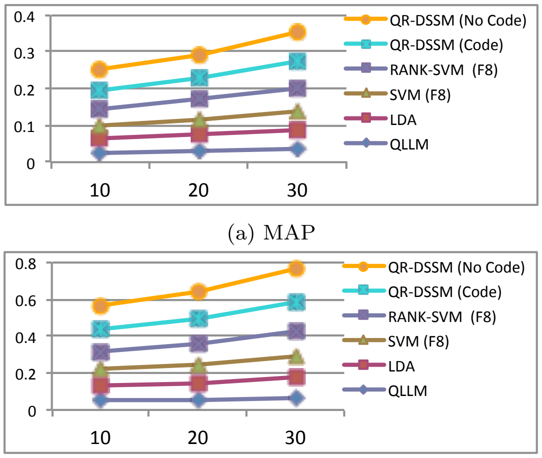
The questions in our dataset are constructed by combining the question title, question body, and question tags. While, the set of users U that provide N answers for the previously extracted training data, 81295 questions, were selected with different N values following the same experiments performed in [11] to ensure a fair comparison.

Training data Q_{Trn} and the testing data Q_{Tst} questions will be asked by an asker, answered by the most skilled answerer and at least one other answerer where they all belong to the U_N set as shown in table 1.

The training questions and the user profiles who answered these questions are combined into pairs. Each question in the training set is paired with every user profile in the training set that provided an answer to that question. Table 2 states the number of pairs generated for each U_N.

Table 3: Comparing the results of the different techniques used for question routing using three datasets (N= 10, 15, 20) where N is the number of questions answered by the user selected in the training and test data. F7 means 7 features are combined in the learning phase

	Metric	LDA	SVM (F7)	Rank-SVM (F7)	QR-DSSM with <code> elements
N=10	MAP	0.0386	0.0363	0.0422	0.05141
	MRR	0.082	0.0847	0.0958	0.1198
N=15	MAP	0.0439	0.443	0.0508	0.0559
	MRR	0.0895	0.0982	0.1103	0.1275
N=20	MAP	0.0527	0.0527	0.0587	0.07499
	MRR	0.1149	0.1149	0.1253	0.1649



(a) MAP

(b) MRR

Figure 3: Comparing question routing results with benchmarks according to the metrics: MAP and MRR

4.3 Evaluation Methodology and Results

The performance of the proposed technique is evaluated and compared against the baseline of Latent Dirichlet Allocation (LDA) [4]. The proposed method is also compared against SVM classifier [5] and the RankingSVM [8].

The evaluation ground truth of the proposed technique is obtained based on the best answerer who actually provided an answer to the test question and the list of the other answerers in U_N. Several evaluation metrics [15][11][3] are widely used for question routing systems such as Mean Reciprocal Rank (MRR) and Mean Average Precision (MAP). The main results of the experiments are summarized in table 3. Figure 4 represents the performance measures for all the experiments question routing methods.

5 CONCLUSION AND FUTUREWORK

In this paper, we proposed a technique (QR-DSSM) for question routing in community question answering (CQA) based on a deep learning technique called Deep Semantic Similarity Model (DSSM). The question routing technique calculates the users' expertise through capturing the similarity between the newly posted question and the users' profiles which were built using the history of questions answered by each user.

Extensive experiments have been conducted on a real-world dataset extracted from Stack Overflow. The results showed that the proposed deep learning technique outperformed the topic modeling technique (LDA), classification technique (SVM) and learning to rank technique RankingSVM in the two metrics used in the experiment MRR and MAP. The best MRR in the experiments is 0.1737 which means on average each test question will get answered if it is routed to the top 6 users.

In future work, we will extend the experiments to compare the results with the QLLM, SVM (F8) and RankSVM-F8. We will also redo the experiments using the same data-set after excluding the code blocks within the textual content. We will also consider two additional metrics such as P@5 and P@10 to assess the quality of the question routing.

6 References

[1] *Learning Deep Structured Semantic Models for Web Search using Clickthrough Data.* ACM International Conference on Information and Knowledge Management (CIKM), October 2013.

[2] *Modeling Interestingness with Deep Neural Networks.* EMNLP, October 2014.

[3] K. Balog, Y. Fang, M. de Rijke, P. Serdyukov, and L. Si. Expertise retrieval. *Found. Trends Inf. Retr.*, 6(2&#8211;3):127–256, Feb. 2012.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[5] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, May 2011.

[6] J. Gao, X. He, and J.-Y. Nie. Clickthrough-based translation models for web search: from word models to phrase models. In *CIKM*, 2010.

[7] R. Herbrich. *Learning Kernel Classifiers: Theory and Algorithms.* MIT Press, Cambridge, MA, USA, 2001.

[8] R. Herbrich, T. Graepel, and K. Obermayer. Support vector learning for ordinal regression. In *In International Conference on Artificial Neural Networks*, pages 97–102, 1999.

[9] A. Hossny, K. Shaalan, and A. Fahmy. Automatic morphological rule induction for arabic. In *Proceedings of the Workshop on Human Language Translation and Natural Language Processing within the Arabic World (LRECâĂŹ08)*, pages 97–101, 2008.

[10] A. Hossny, K. Shaalan, and A. Fahmy. Machine translation model using inductive logic programming. In *Natural Language Processing and Knowledge Engineering, 2009. NLP-KE 2009. International Conference on*, pages 1–8. IEEE, 2009.

[11] Z. Ji and B. Wang. Learning to rank for question routing in community question answering. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 2363–2368, New York, NY, USA, 2013. ACM.

[12] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, Sept. 1999.

[13] D.-R. Liu, Y.-H. Chen, W.-C. Kao, and H.-W. Wang. Integrating expert profile, reputation and link analysis for expert finding in question-answering websites. *Inf. Process. Manage.*, 49(1):312–329, 2013.

[14] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[15] F. Riahi, Z. Zolaktaf, M. Shafiei, and E. Milios. Finding expert users in community question answering. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12 Companion, pages 791–798, New York, NY, USA, 2012. ACM.

[16] R. Salakhutdinov and G. Hinton. Semantic hashing. *Int. J. Approx. Reasoning*, 50(7):969–978, July 2009.

[17] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523, Aug. 1988.

[18] Y. Zhou, G. Cong, B. Cui, C. S. Jensen, and J. Yao. Routing questions to the right users in online communities. In *Proceedings of the 2009 IEEE International Conference on Data Engineering*, ICDE '09, pages 700–711, Washington, DC, USA, 2009. IEEE Computer Society.