

# An Interactive System for Exploring Community Question Answering Forums

Enamul Hoque<sup>†‡</sup>, Shafiq Joty<sup>†</sup>, Lluís Màrquez<sup>†</sup>, Alberto Barrón-Cedeño<sup>†</sup>, Giovanni Da San Martino<sup>†</sup>  
Alessandro Moschitti<sup>†</sup>, Preslav Nakov<sup>†</sup>, Salvatore Romeo<sup>†</sup>, and Giuseppe Carenini<sup>‡</sup>

<sup>†</sup>ALT group, Qatar Computing Research Institute, HBKU, Qatar Foundation

<sup>‡</sup>Department of Computer Science, University of British Columbia

<sup>†</sup>{`sjoty, lmarquez, albarron, amoschitti, gmartino, pnakov, sromeo`}@qf.org.qa

<sup>‡</sup>{`enamul, carenini`}@cs.ubc.ca

## Abstract

We present an interactive system to provide effective and efficient search capabilities in Community Question Answering (cQA) forums. The system integrates state-of-the-art technology for answer search with a Web-based user interface specifically tailored to support the cQA forum readers. The answer search module automatically finds relevant answers for a new question by exploring related questions and the comments within their threads. The graphical user interface presents the search results and supports the exploration of related information. The system is running live as a part of the Qatar Living forums.

## 1 Introduction

Community Question Answering (cQA) forums, such as StackOverflow and Quora, are becoming more and more popular these days.<sup>1</sup> They represent effective means for communities of users around particular topics to share information and to collectively solve their information needs. cQA forums typically organize their content in the form of multiple topic-oriented *question–comment threads*, where a question posed by a user may be answered by a possibly very long list of other users' comments.

Many such on-line forums are not moderated, which often results in noisy and redundant content. Users tend to initiate new questions or engage in discussions that easily deviate from the original topic. Additionally, the same questions may be posted repeatedly with minor variations. This near-duplication is very difficult to track for users, who are usually offered simple search capabilities by the forum interface. Finding existing good answers to newly-posed questions (i.e., never asked in exactly this way before) is a real challenge for cQA, since they may be scattered around multiple related conversations and buried among a large number of comments. Recently, automatic systems have been proposed to address this problem in the framework of the SemEval-2015 and SemEval-2016 tasks on cQA (Nakov et al., 2015; Nakov et al., 2016).<sup>2</sup>

In this paper, we present an interactive system tailored to help users to find good answers to a new question and we apply it to the Qatar Living forum. The system integrates search and NLP modules to (i) find related questions in the forum, and (ii) rank by relevance the comments within the thread for each such related question. The top suggested answer to the original question is found by a combination of these two processes. The core NLP part of our system is the answer ranking module. This is an improved version of the state-of-the-art classifier with which we participated in SemEval-2016 Task 3 (Barrón-Cedeño et al., 2016).

Our system integrates a Web-based interface to address the further challenges that arise in presenting the results to the user. The interface allows the user to start with a new question, then to explore the related threads to find the ones that are most relevant to his/her information needs, and eventually to navigate through the comments of a thread looking for relevant answers to the question.

<sup>1</sup><http://stackoverflow.com>, <https://www.quora.com>

<sup>2</sup><http://alt.qcri.org/semEval\{2015,2016\}/task3/>

This work is licensed under a Creative Commons Attribution 4.0 International License. License details can be found here: <http://creativecommons.org/licenses/by/4.0/>

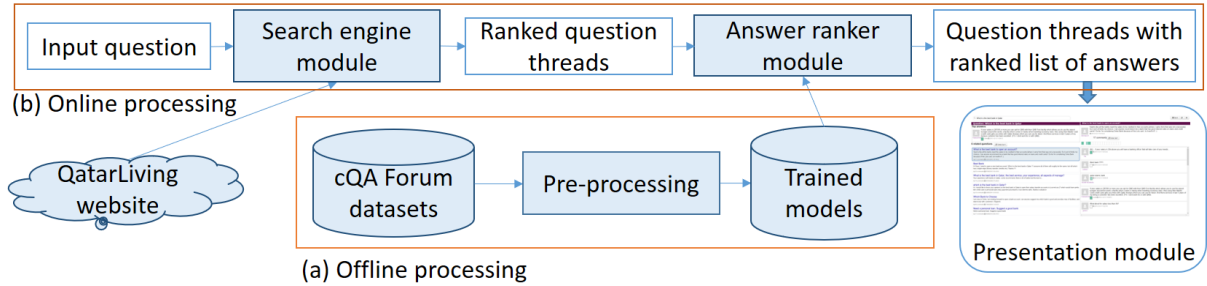


Figure 1: Overview of our interactive system for supporting community question answering.

## 2 System Overview

An overview of our system is shown in Figure 1. We first perform some offline steps to process the data and to train the rerankers (Subfigure a). The proper on-line system is illustrated in Subfigure b. In the remainder of this section, we briefly discuss these steps.

**Offline Processing** In order to build the system, we obtained a recent dump of the Qatar Living forum (from March 2016), and we performed several formatting pre-processing steps. We also used the cQA dataset from SemEval-2016 Task 3 (subtask A), where the comments in the threads are annotated with `good` vs. `bad` labels indicating how well the comments answer the question in the thread. Using this dataset, we extracted features and we trained a kernel-based comment classifier (cf. Section 3). The trained models are used to provide goodness scores for each comment in each thread.

**Online Processing** When a user types a new question  $q$ , the system performs the following three steps on the fly: (i) *Retrieving related questions with a search engine module*, where Google local search is invoked to retrieve the top- $n$  question threads in the Qatar Living forum that are most similar to  $q$ ; (ii) *Ranking the answers*, where all the comments from these top- $n$  question threads are ranked based on their relevance with respect to  $q$  (see Section 3 for detail); (iii) *Visualizing the results*, where the presentation module takes the related questions’ threads together with the ranked lists of comments and the overall best selected answer, and presents them to the user within an interactive Web interface (see Section 4 below).

## 3 Ranking Answers with Respect to the Input Question

We compute the relevance score of a comment  $c$  in a question thread  $q'$  with respect to the original question  $q$  by multiplying: (i) the relevance of  $q'$  to  $q$  (we use the inverse rank in the list returned by the Google search engine) by (ii) the goodness score for  $c$  with respect to  $q'$  (produced by the comment classifier, and indicating how well comment  $c$  answers  $q'$ ). The resulting score is used to rank all the comments from the retrieved question threads to obtain the best overall answer to the input question. The core NLP component of this architecture is the comment classifier, which is briefly described below.

**The Comment Classifier** Given a question and a set of comments associated with it, the task is to assign a relevance score to each of the comments according to their *goodness* at answering the question. This very problem was set at SemEval-2016 Task 3 (Nakov et al., 2016). We trained a Support Vector Machine (SVM) classifier on the SemEval-2016 subtask A dataset to distinguish between `good` and `bad` comments. The kernel function in our SVM is a linear combination of four functions: two linear kernels over numeric features and embeddings, and two tree kernels over shallow syntactic trees.

*Numeric Features* They include three types of information: (i) a variety of textual similarity measures computed between the question and the comment, (ii) several Boolean features capturing the presence of URLs, emails, positive/negative words, acknowledgments, forum categories, long words, etc., and (iii) a set of global features modeling dialogue and user interactions in the thread. More detailed descriptions of these features can be found in (Barrón-Cedeño et al., 2015; Nicosia et al., 2015; Joty et al., 2015).

*Embedding Features* We learn embeddings for questions and answers by training a convolutional neural network (CNN) on the comment classification task following (Severyn and Moschitti, 2015). Specifically, the input to the CNN is formed by two matrices containing word embeddings for the question and for the answer, respectively. The CNN performs a *convolution* and a *max-pooling* operations on the word embeddings and on the convoluted feature maps, respectively, to produce the question embedding  $q_e$  and the answer embedding  $c_e$ . These embeddings are then combined to produce a similarity value using a similarity matrix. The similarity and the embeddings along with other additional similarity features are then passed through a hidden layer and next to the output layer for classification. The  $q_e$  and  $c_e$  are learned by backpropagating the (cross entropy) errors from the output layer.  $q_e$  and  $c_e$  vectors are finally concatenated and used as features in our SVM model.

*Tree kernels* We use tree kernels to measure the syntactic similarity between the question and the comment. First, we produce shallow syntactic trees for the question and for the comment using the Stanford parser (Klein and Manning, 2003). Following Severyn and Moschitti (2012), we link the two trees by connecting nodes such as NP, PP, VP, when there is at least one lexical overlap between the corresponding phrases of the trees, and we mark those links using a specific tag. The kernel function  $K$  is defined as:  $K((t_1, t_2), (c_1, c_2)) = TK(t_1, c_1) + TK(t_2, c_2)$ , where  $TK(t, c)$  is a tree kernel function operating over a pair of question ( $t$ ) and comment ( $c$ ) trees.<sup>3</sup>

*Classification Performance* We evaluated our comment classifier on the SemEval-2016 Task 3 test set with the official scorer, obtaining the following results: MAP=77.66, AvgRec=88.05, MRR=84.93,  $F_1$ =66.16, Acc=75.54. Compared to the systems that took part in the competition, our system would have ranked in second position according to the official MAP evaluation metric (−1.5 points below the best). In contrast, we achieve better  $F_1$  (+1.8) and better Accuracy (+0.4) than the top system. For a full comparison to the SemEval-2016 Task 3 results see (Nakov et al., 2016).

## 4 The System in Action

The design of our visual interface was guided by previous research on designing interfaces for exploring online conversations (Hoque and Carenini, 2016); however, in this new design we took into account specific features of cQA data and tasks. Our interface consists of the following components: a search bar, a *question list view* that shows the top-most relevant questions to the user’s question; and a *conversation view* showing the question followed by the answers for a particular question thread (see Figure 2).

**Questions list view:** After the system finds the related questions to the user’s question, it presents the top relevant questions in a scrollable list view (see Figure 2, left). Each item within the question list view represents a question thread, showing the original question, the posting date, and a stacked bar with the distribution of useful comments. In this way, the user can get a sense of which threads seem to be more relevant and which threads may contain the most useful answers. The questions are ordered by their relevance rank by default, but the user can change this order by selecting criteria from the popup menu ‘Order by’. For instance, s/he can order the question threads based on the number of useful answers within each of these threads. Finally, at any time, the user can filter out less useful comments by using the slider of the legend at the top. Note that on top of the question list view, the interface also shows the comment that has received the best score with respect to the new question (“Best Answer”). This way, the user may be able to find a very good answer to his/her question immediately, without having to open any question thread and then navigating to a good answer within that thread.

**Conversation view:** When the user selects a particular question thread from the list, the system presents the corresponding thread in the conversation view (see Figure 2, right). On top of this conversation view, the original question along with a visual overview of the entire thread is presented, followed by the list of detailed comments.<sup>4</sup> The thread overview visually encodes the comments using a sequence of rectangles from left to right, where each rectangle represents a comment. A set of five sequential colors was used in a perceptually meaningful order, ranging from dark green (highly useful) to white (not useful) to encode the classification score for each comment.

<sup>3</sup>We use Partial Tree Kernel and Syntactic Tree Kernel (Moschitti, 2006; Collins and Duffy, 2001) to instantiate  $TK$ .

<sup>4</sup>Note that the red rectangle in Figure 2 is only used to highlight the thread overview; it is not displayed in the real interface.

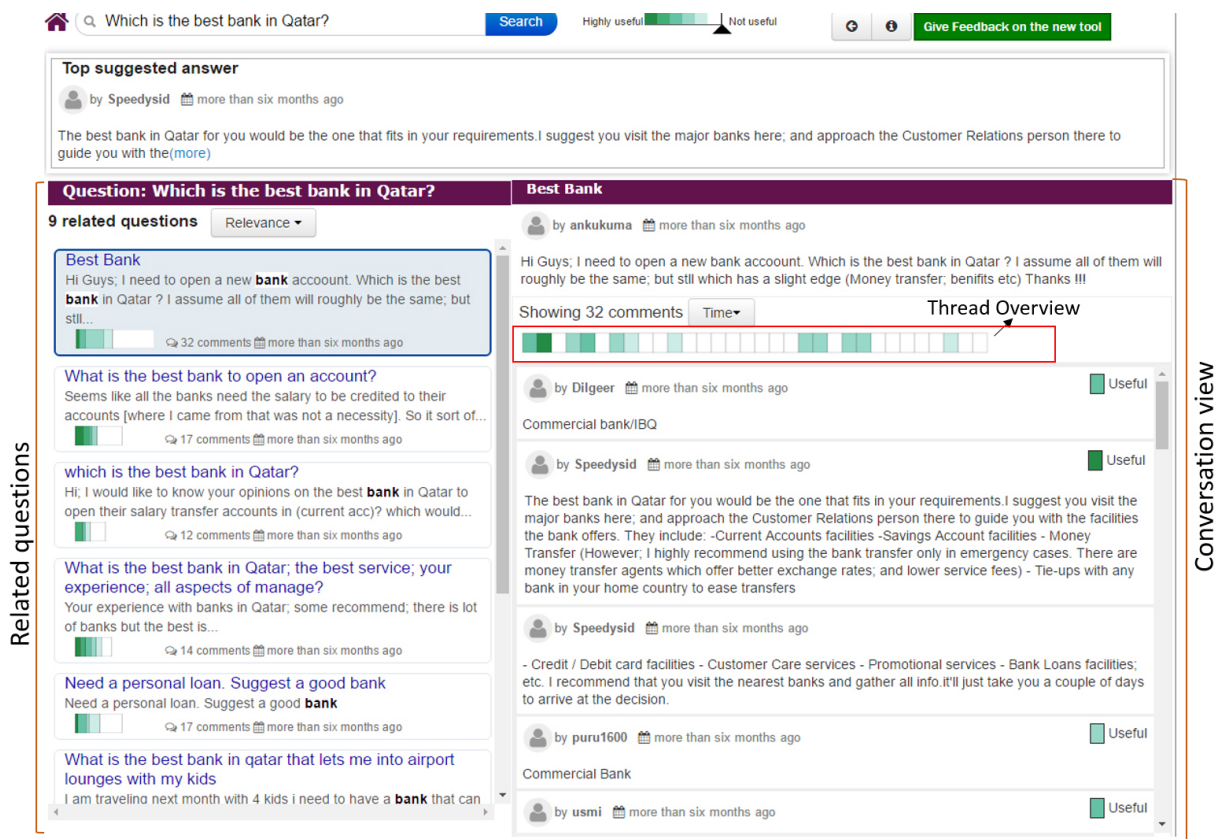


Figure 2: A screenshot of the interface showing the top answer and related questions for a user’s question. As the user selects a related question (marked by the blue rectangular boundary), the interface shows the corresponding thread in the conversation view (right).

From the thread overview, the user can quickly notice which comments seem to be more useful and then can immediately navigate to a particular comment by clicking on the rectangle representing that comment. Note that hovering on a rectangle in the thread view highlights the corresponding comment in the detailed view (by scrolling if needed) and vice-versa.

**Implementation** The system is implemented as a Java Web application and runs on an Apache Tomcat Server. The back-end of the system is developed using Java. The presentation module, on the other hand, is implemented in Javascript (using the D3 and JQuery libraries). The system is sufficiently fast to respond in real time to the user’s actions. A key factor for the efficiency is the fact that we precomputed and stored the goodness scores for all the comments in all the question-threads from the static snapshot of the Qatar Living database. Thus, at running time there is no need to classify the comments of the already stored question-comment threads.

## 5 Conclusion

We have presented an interactive system that supports users to find good answers to newly-posed questions using pre-existing questions and their answer threads in community question answering forums. In particular, we implemented a Web-based demo trained on data from SemEval-2016 Task 3 and allows users to ask questions and to get real-time answers using data from the Qatar Living forum. The demo has already been deployed in Qatar Living.<sup>5</sup> It provides a graphical interface, which allows users to navigate in the set of related questions (question-list view) and in the set of comments in a thread (conversation view). Internally, the system uses state-of-the-art NLP tools and search capabilities to effectively retrieve and rerank a set of comments with respect to the new question.

<sup>5</sup><http://www.qatarliving.com/betasearch/>

In future work, we will evaluate the demo interface by running user studies with real Qatar Living users. We also plan to further improve all the classifiers of our system.

## Acknowledgments

This research was performed by the Arabic Language Technologies (ALT) group at the Qatar Computing Research Institute (QCRI), HBKU, Qatar Foundation. It is part of the Interactive sYstems for Answer Search (Iyas) project, which is developed in collaboration with MIT-CSAIL. We thank Scott Cyphers and Mitra Mohtarami for their help in designing and implementing the initial demo architecture.

## References

- Alberto Barrón-Cedeño, Simone Filice, Giovanni Da San Martino, Shafiq Joty, Lluís Màrquez, Preslav Nakov, and Alessandro Moschitti. 2015. Thread-level information for comment classification 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 (Volume 2: Short Papers)*, ACL-IJCNLP '15, pages 687–693, Beijing, China.
- Alberto Barrón-Cedeño, Giovanni Da San Martino, Shafiq Joty, Alessandro Moschitti, Fahad Al-Obaidli, Salvatore Romeo, Kateryna Tymoshenko, and Antonio Uva. 2016. ConvKN at SemEval-2016 Task 3: Answer and question selection for question answering on Arabic and English fora. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, pages 896–903, San Diego, California, USA.
- Michael Collins and Nigel Duffy. 2001. Convolution kernels for natural language. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Advances in Neural Information Processing Systems*, NIPS '01, pages 625–632, Vancouver, British Columbia, Canada. MIT Press.
- Enamul Hoque and Giuseppe Carenini. 2016. MultiConVis: A Visual Text Analytics System for Exploring a Collection of Online Conversations. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, IUI '2016, pages 96–107, Sonoma, California, USA.
- Shafiq Joty, Alberto Barrón-Cedeño, Giovanni Da San Martino, Simone Filice, Lluís Màrquez, Alessandro Moschitti, and Preslav Nakov. 2015. Global thread-level inference for comment classification in community question answering. In *Proc. EMNLP*, pages 573–578.
- Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 423–430, Sapporo, Japan.
- Alessandro Moschitti. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. In *Proceedings of the 17th European Conference on Machine Learning*, ECML'06, pages 318–329, Berlin, Germany.
- Preslav Nakov, Lluís Màrquez, Walid Magdy, Alessandro Moschitti, Jim Glass, and Bilal Randeree. 2015. Semeval-2015 task 3: Answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '15, pages 269–281, Denver, Colorado, USA.
- Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016. SemEval-2016 Task 3: Community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, pages 525–545, San Diego, California, USA.
- Massimo Nicosia, Simone Filice, Alberto Barrón-Cedeño, Iman Saleh, Hamdy Mubarak, Wei Gao, Preslav Nakov, Giovanni Da San Martino, Alessandro Moschitti, Kareem Darwish, Lluís Màrquez, Shafiq Joty, and Walid Magdy. 2015. QCRI: Answer selection for community question answering - experiments for Arabic and English. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '15, pages 203–209, Denver, Colorado, USA.
- Aliaksei Severyn and Alessandro Moschitti. 2012. Structural relationships for large-scale learning of answer re-ranking. In *Proceedings of the 35th International Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 741–750, Portland, Oregon, USA.
- Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 373–382, Santiago, Chile.