Thread-Level Information for Comment Classification in Community Question Answering

Alberto Barrón-Cedeño, Simone Filice, Giovanni Da San Martino, Shafiq Joty, Lluís Màrquez, Preslav Nakov, Alessandro Moschitti
Qatar Computing Research Institute, Hamad Bin Khalifa University
{albarron,sfilice,gmartino, sjoty,lmarquez, pnakov,amoschitti}@qf.org.qa

Abstract

Community Question Answering (cQA) is a new application of QA in social contexts (e.g., fora). It presents new interesting challenges and research directions, e.g., exploiting the dependencies between the different comments of a thread to select the best answer for a given question. In this paper, we explored two ways of modeling such dependencies: (*i*) by designing specific features looking globally at the thread; and (*ii*) by applying structure prediction models. We trained and evaluated our models on data from SemEval-2015 Task 3 on Answer Selection in cQA. Our experiments show that: (*i*) the thread-level features consistently improve the performance for a variety of machine learning models, yielding state-of-the-art results; and (*ii*) sequential dependencies between the answer labels captured by structured prediction models are not enough to improve the results, indicating that more information is needed in the joint model.

1 Introduction

Community Question Answering (cQA) is an evolution of a typical QA setting put in a Web forum context, where user interaction is enabled, without much restrictions on who can post and who can answer a question. This is a powerful mechanism, which allows users to freely ask questions and expect some good, honest answers.

Unfortunately, a user has to go through all possible answers and to make sense of them. It is often the case that many answers are only loosely related to the actual question, and some even change the topic. This is especially common for long threads where, as the thread progresses, users start talking to each other, instead of trying to answer the initial question.

This is a real problem, as a question can have hundreds of answers, the vast majority of which would not satisfy the users' information needs. Thus, finding the desired information in a long list of answers might be very time-consuming.

The problem of selecting the relevant text passages (i.e., those containing good answers) has been tackled in QA research, either for non-factoid QA or for passage reranking. Usually, automatic classifiers are applied to the answer passages retrieved by a search engine to derive a relative order; see (Radlinski and Joachims, 2005; Jeon et al., 2005; Shen and Lapata, 2007; Moschitti et al., 2007; Surdeanu et al., 2008; Heilman and Smith, 2010; Wang and Manning, 2010; Severyn and Moschitti, 2012; Yao et al., 2013; Severyn et al., 2013; Severyn and Moschitti, 2013) for detail.

To the best of our knowledge, there is no QA work that effectively identifies good answers based on the selection of the other answers retrieved for a question. This is mainly due to the loose dependencies between the different answer passages in standard QA. In contrast, we postulate that in a cQA setting, the answers from different users in a common thread are strongly interconnected and, thus, a joint answer selection model should be adopted to achieve higher accuracy.

To test our hypothesis about the usefulness of thread-level information, we used a publicly available dataset, recently developed for the SemEval-2015 Task 3 (Nakov et al., 2015). Subtask A in that challenge asks to identify the posts in the answer thread that answer the question *well* vs. those that can be *potentially useful* to the user vs. those that are just *bad or useless*.

We model the thread-level dependencies in two different ways: (*i*) by designing specific features that are able to capture the dependencies between the answers in the same thread; and (*ii*) by exploiting the sequential organization of the output labels for the complete thread.

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers), pages 687–693, Beijing, China, July 26-31, 2015.

Q: Can I obtain Driving License my QID is written Employee

$\mathbf{A}_1$ the word employee is a general term that refers to all the staff in your company either the manager, secretary up to the lowest position or whatever positions they have. you are all considered employees of your company.

$\mathbf{A}_2$ your qid should specify what is the actual profession you have. i think for me, your chances to have a drivers license is low.

$\mathbf{A}_3$ dear richard, his asking if he can obtain. means he have the driver license

$\mathbf{A}_4$ Slim chance ...

Figure 1: Simplified example from SemEval-2015 Task 3, English subtask A.

For the latter, we used the usual extensions of Logistic Regression and SVM to linear-chain models such as CRF and SVM^{hmm}.

The results clearly show that the thread-level features are important, providing consistent improvement for all our learning models. In contrast, the linear-chain models fail to exploit the sequential dependencies between nearby answer labels to improve the results significantly: although the labels from the neighboring answers can affect the label of the current answer, this dependency is too loose to have impact on the selection accuracy. In other words, labels should be used together with answers' content to account for stronger and more effective dependencies.

2 The Task

We use the CQA-QL corpus, which was used for Subtask A of SemEval-2015 Task 3 on Answer Selection in cQA. The corpus contains data from the *Qatar Living* forum,[1] and is publicly available on the task's website.[2] The dataset consists of questions and a list of the answers for each question, i.e., the *question-answer thread*. Each question, and also each answer, consists of a short title and a more detailed description. Moreover, there is some meta information associated with both, e.g., ID of the user asking/answering the question, timestamp, question category, etc.

The task asks to determine for each answer in the thread whether it is good, bad, or potentially useful. A simplified example is shown in Figure 1,[3] where answers 2 and 4 are good, answer 1 is potentially useful, and answer 3 is bad.

[1] http://www.qatarliving.com/forum

[2] http://alt.qcri.org/semeval2015/task3/

[3] http://www.qatarliving.com/moving-qatar/posts/can-i-obtain-driving-license-my-qid-written-employee

Below, we start with the original definition of Subtask A, as described above. Then, we switch to a binary classification setting (i.e., identifying *good* vs. *bad* answers), which is much closer to a real cQA application (see Section 4.3).

3 Basic and Thread-Level Features

Subsection 3.1 summarizes the basic features we used to implement the baseline systems. More importantly, Section 3.2 describes the set of thread-level features we designed in order to test our working hypothesis. Below we use the following notation: q is a question posted by user u_q, c is a comment, and C is the comment thread.

3.1 Baseline Features

We measure lexical and syntactic similarity between q and c. We compute the similarity between word n-grams ($n = [1, \ldots, 4]$), after stopword removal, using greedy string tiling (Wise, 1996), longest common subsequences (Allison and Dix, 1986), Jaccard coefficient (Jaccard, 1901), word containment (Lyon et al., 2001), and cosine similarity. We also apply partial tree kernels (Moschitti, 2006) on shallow syntactic trees.

We designed a set of heuristic features that might suggest whether c is *good* or not. Forty-four Boolean features express whether c (*i*) includes URLs or emails (2 feats.); (*ii*) contains the word "yes", "sure", "no", "can", "neither", "okay", and "sorry", as well as symbols '?' and '@' (9 feats.); (*iii*) starts with "yes" (1 feat.); (*iv*) includes a sequence of three or more repeated characters or a word longer than fifteen characters (2 feats.); (*v*) belongs to one of the categories of the forum (*Socialising*, *Life in Qatar*, etc.) (26 feats.); and (*vi*) has been posted by the same u_q, such a comment can include a question (i.e., contain a question mark), and acknowledgement (e.g., contain *thank**, *acknowl**), or none of them (4 feats.). An extra feature captures the length of c (as longer — *good*— comments usually contain detailed information to answer a question).

3.2 Thread-Level Global Features

Comments are organized sequentially according to the time line of the comment thread.[4] Our first four features indicate whether c appears in the proximity of a comment by u_q.

[4] The task organizers report that some comments in the threads were discarded due to disagreement in the annotation process. The extent of discarded comments is unknown.

	P_{ca}	R_{ca}	$F_{1,ca}$	A
Baseline Features				
SVM	52.96	53.14	52.87	67.61
OrdReg	53.33	51.54	51.87	65.38
Baseline+Thread-level Features				
SVM	56.31	56.46	56.33	72.27
OrdReg	57.68	57.04	57.20	72.47
SemEval top three				
JAIST	57.31	57.20	57.19	72.52
HITSZ	57.83	56.82	56.41	68.67
QCRI	54.34	53.57	53.74	70.50

Table 1: Macro-averaged precision, recall, F_1-measure, and accuracy on the multi-class (*good, bad, potential*) setting on the official SemEval-2015 Task 3 test set. The top-2 systems are included for comparison. QCRI refers to our official results, using an older version of our system.

The assumption is that an acknowledgment or further questions by u_q in the thread could signal a *good* answer. More specifically, they test if among the comments following c there is one by u_q (*i*) containing an acknowledgment, (*ii*) not containing an acknowledgment, (*iii*) containing a question, and, (*iv*) if among the comments preceding c there is one by u_q containing a question. The value of these four features —a propagation of the information captured by some of the heuristics described in Section 3.1— depends on the distance k, in terms of the number of comments, between c and the closest comment by u_q:

$$f(c) = \begin{cases} \max(0,\ 1.1 - (k \cdot 0.1)) \\ 0 \text{ if no comments by } u_q \text{ exist,} \end{cases} \quad (1)$$

that is, the closer the comment to c_q, the higher the value assigned to this feature.

We try to model potential dialogues, which at the end represent *bad* comments, by identifying interlacing comments between two users. Our dialogue features are identifying conversation chains: $u_i \rightarrow \ldots \rightarrow u_j \rightarrow \ldots \rightarrow u_i \rightarrow \ldots \rightarrow [u_j]$. Comments by other users can appear in between the nodes of this "pseudo-conversation" chain. We consider three features: whether a comment is at the beginning, in the middle, or at the end of such a chain. Three more features exist in those cases in which u_q is one of the participants of these pseudo-conversations.

Another Boolean feature for c_{u_i} is set to true if u_i wrote more than one comment in the current thread. Three more features identify the first, the middle and the last comments by u_i. One extra feature counts the total number of comments written by u_i in the thread up to that moment.

	P	R	F_1	A	$F_{1,ta}$	A_{ta}
Baseline Features						
SVM	70.58	84.45	76.89	74.39	66.52	76.13
SVM^{hmm}	72.57	85.46	78.49	76.37	68.55	77.58
LogReg	65.05	91.27	75.96	70.85	68.84	74.79
CRF_{map}	72.48	86.66	78.94	76.67	67.17	76.55
CRF_{mpm}	71.55	84.25	77.38	75.15	66.54	75.42
Baseline+Thread-level Features						
SVM	75.29	85.26	79.96	78.44	67.65	76.02
SVM^{hmm}	74.84	83.25	78.82	77.43	66.61	77.06
LogReg	73.32	86.56	79.39	77.33	68.10	75.57
CRF_{map}	73.77	85.76	79.31	77.43	66.37	76.08
CRF_{mpm}	74.35	85.46	79.51	77.78	67.36	76.63

Table 2: Performance of the binary (*good* vs. *bad*) classifiers on the official SemEval-2015 Task 3 test dataset. Precision, recall, F_1-measure and accuracy are calculated at the comment level, while $F_{1,ta}$ and A_{ta} are averaged at the thread level.

Moreover, we empirically observed that the likelihood of some comment being *good* decreases with its position in the thread. Therefore, we also included another real-valued feature: $\max(20, i)/20$, where i represents the position of the comment in the thread.

Finally, we perform a pseudo-ranking of the comments. The relevance of c is computed as its similarity to q (using word n-grams), normalized by the maximum similarity among all the comments in the thread. The resulting relative scores are mapped into three binary features depending on the range they fall at: $[0, 0.2]$, $(0.2, 0.8)$, or $[0.8, 1]$ (intervals resemble the three-class setting and were empirically set on the training data).

4 Experiments

Below we first describe the data we used, then we introduce the experimental setup, and finally we present and discuss the results of our experiments.

4.1 Data

The original CQA-QL corpus (Nakov et al., 2015) consists of 3,229 questions: 2,600 for training, 300 for development, and 329 for testing. The total number of comments is 20,162, with an average of 6.24 comments per question. The class labels for the comments are distributed as follows: 9,941 *good* (49.31%), 2,013 *potential* (9.98%), and 8,208 *bad* (40.71%) comments.

Since a typical answer selection setting only considers correct and incorrect answers, we also experiment with *potential* labelled as *bad*.

	P	R	F_1	A	$F_{1,ta}$	A_{ta}
Baseline Features						
SVM	68.86±1.42	82.34±1.04	74.98±0.73	72.90±1.00	64.56±0.97	75.32±0.40
SVM^{hmm}	70.34±1.57	81.00±1.98	75.28±1.05	73.75±1.56	65.25±1.16	74.68±1.05
LogReg	64.20±1.33	88.54±0.81	74.42±0.80	69.99±0.94	66.00±1.33	73.04±0.96
CRF_{map}	69.11±1.41	80.63±1.76	74.42±1.29	72.66±1.75	63.90±1.71	73.51±0.73
CRF_{mpm}	69.60±1.65	81.17±1.28	74.93±1.19	73.20±1.77	64.53±1.37	74.32±0.92
Baseline+Thread-level Features						
SVM	72.55±0.96	83.39±1.36	77.59±0.95	76.23±1.37	66.41±1.30	76.23±0.45
SVM^{hmm}	73.24±1.66	81.66±1.21	77.21±1.18	76.20±1.81	65.33±1.12	76.43±0.92
LogReg	71.15±0.96	84.44±1.50	77.22±1.07	75.43±1.47	66.57±1.49	75.05±0.70
CRF_{map}	71.27±1.20	83.15±1.81	76.75±1.28	75.14±1.72	65.36±1.45	75.61±0.63
CRF_{mpm}	71.56±1.31	83.34±1.84	77.00±1.35	75.43±1.84	65.57±1.54	75.71±0.71

Table 3: Precision, Recall, F_1, Accuracy computed at the comment level; $F_{1,ta}$ and A_{ta} are averaged at the thread level. Precision, Recall, F_1, $F_{1,ta}$ are computed with respect to the *good* classifier on 5-fold cross-validation (mean±stand. dev.).

4.2 Experimental Setup

Our local classifiers are support vector machines (SVM) with $C = 1$ (Joachims, 1999), logistic regression with a Gaussian prior with variance 10, and logistic ordinal regression (McCullagh, 1980). In order to capture long-range sequential dependencies, we use a second-order SVM^{hmm} (Yu and Joachims, 2008) (with $C = 500$ and $epsilon = 0.01$) and a second-order linear-chain CRF, which considers dependencies between three neighboring labels in a sequence (Lafferty et al., 2001; Cuong et al., 2014). In CRF, we perform two kinds of inference to find the most probable labels for the comments in a sequence. (*i*) We compute the maximum a posterior (MAP) or the (jointly) most probable sequence of labels using the Viterbi algorithm. Specifically, it computes $\mathbf{y}^* = \texttt{argmax}_{\mathbf{y}_{1:T}} P(\mathbf{y}_{1:T}|\mathbf{x}_{1:T})$, where T is the number of comments in the thread. (*ii*) We use the forward–backward algorithm to find the labels by maximizing (individual) posterior marginals (MPM). More formally, we compute $\hat{\mathbf{y}} = \big(\texttt{argmax}_{\mathrm{y}_1} P(y_1|\mathbf{x}_{1:T}), \cdots, \texttt{argmax}_{\mathrm{y}_T} P(y_T|\mathbf{x}_{1:T})\big)$. While MAP yields a globally consistent sequence of labels, MPM can be more robust in many cases; see (Murphy, 2012, p. 613) for details. CRF also uses a Gaussian prior with variance 10.[5]

4.3 Experiment results

In order to compare the quality of our features to the existing state of the art, we performed a first experiment aligned to the multi-class setting of the SemEval 2015 Task 3 competition. Table 1 shows our results on the official test dataset.

[5]Varying regularization strength (variance of the prior) did not make much difference.

As in the competition, the results are macro-averaged at class level. The results of the top 3 systems are reported for comparison: JAIST (Tran et al., 2015), HITSZ (Hou et al., 2015) and QCRI (Nicosia et al., 2015), where the latter refers to our old system that we used for the competition. The two main observations are (*i*) using thread-level features helps significantly; and (*ii*) the ordinal regression model, which captures the idea that *potential* lies between *good* and *bad*, achieves at least as good results as the top system at SemEval, namely JAIST.

For the remaining experiments, we reduce the multi-class problem to a binary one (cf. Section 2). Table 2 shows the results obtained on the official test dataset. Note that ordinal regression is not applicable in this binary setting. The F_1 values for the baseline features suggest that using the labels in the thread sequence yields better performance with SVM^{hmm} and CRF. When thread-level features are used, the models using sequence labels do not outperform SVM and logistic regression anymore. Regarding the two variations of CRF, the posterior marginals maximization is slightly better: maximizing on each comment pays more than on the entire thread.

Since the task consists in identifying *good* answers for a given question, further figures at the question level are necessary, i.e., we compute the target performance measure for all comments of each question and then we average the results over all threads (ta). Table 2 shows such the result using two measures: F_1 and accuracy, i.e., $F_{1,ta}$ and A_{ta}, for which long threads have less impact on the final outcome. The impact of the thread features is not-so-high in terms of these measures, sometimes even negatively affecting some of the models.

Q_{u_1}:	Gymnastic world cup. Does anyone know what time the competition starts today? Thanks	
c_{1,u_2}:	sorry - is this being held here in Doha? If so, I'd love to go. Expat Sueo P.S. Is that a labradoodle in your avatar?	Bad→Bad
c_{2,u_1}:	No actually a Cockapoo! Yes the comp. runs from today until Wednesday at Aspire.	Bad→Bad
c_{3,u_2}:	Thanks for the info - maybe I'll turn up after the TableTop Sale is done and dusted! ES P.S. Cute pup!	Good→Bad
Q_{u_4}:	Good Scissor. Dears, anyone have an idea where to find a good scissor for hair and beard trimming please???	
c_{1,u_5}:	Visit Family food center	Bad→Good
c_{2,u_6}:	Al rawnaq airport road...U'll find all types of scissors there...	Bad→Good
c_{3,u_4}:	Thank you all . . . I will try that.	Bad→Bad

Figure 2: Two real question–comments threads (simplified; ID in CQA-QL: Q770 and Q752). The sub-indexes stand for the position in the thread and the author of the comment. The class label corresponds to the prediction before and after considering thread-level information. The right-hand label matches with the gold one in all the cases.

Cross validation. In order to better understand the mixed results obtained on the single official test set, we performed 5-fold cross validation over the entire dataset. The results are shown in Table 3. When looking at the performance of the different models with the same set of features, no statistically significant differences are observed on F_1 or $F_{1,ta}$ (t-test with confidence level 95%). The sequence of predicted labels in CRF or SVM^{hmm} does not impact the final result. In contrast, an important difference is observed when thread-level features come into play: the performance of all the models improves by approximately two F_1 points absolute, and statistically significant differences are observed for SVM and logistic regression (t-test, 95%). Moreover, while on the test dataset the thread-level features do not always improve $F_{1,ta}$ and A_{ta}, on the 5-fold cross-validation using them is always beneficial: for $F_{1,ta}$ statistically significant difference is observed for SVM only (t-test, 90%).

Qualitative results. In order to get an intuition about the effect of the thread-level features, we show two example comment threads in Figure 2. These comments are classified correctly when thread features are used in the classifier, and incorrectly when only basic features are used.

In the first case (Q_{u_1}), the third comment is classified as *good* by models that only use basic features. In contrast, thanks to the thread-level features, the classifier can consider that there is a dialogue between u_1 and u_2, causing all the comments to be assigned to the correct class: *bad*.

In the second example (Q_{u_4}), the first two comments are classified as *bad* when using the basic features. However, the third comment —written by the same user who asked Q_{u_4}— includes an acknowledgment. The latter is propagated to the previous comments in terms of a thread feature, which indicates that such comments are more likely to be *good* answers. This feature provides the classifier with enough information to properly label the first two comments as *good*.

5 Conclusions

We presented a study on using dependencies between the different answers in the same question thread in the context of answer selection in cQA. Our experiments with different classifiers, features, and experimental conditions, reveal that answer dependencies are helpful to improve results on the task. Such dependencies are best exploited by means of carefully designed thread-level features, whereas sequence label information alone, e.g., used in CRF or SVM^{hmm}, is not effective.

In future work, we plan to (*i*) experiment with more sophisticated thread-level features, as well as with other features that model context in general; (*ii*) try data from other cQA websites, e.g., where dialogue between users is marked explicitly; and finally, (*iii*) integrate sequence, precedence, dependency information with global —thread-level— features in a unified framework.

Acknowledgments

This research is developed by the Arabic Language Technologies (ALT) group at the Qatar Computing Research Institute (QCRI), Hamad Bin Khalifa University, within Qatar Foundation in collaboration with MIT. It is part of the Interactive sYstems for Answer Search (Iyas) project.

References

Lloyd Allison and Trevor Dix. 1986. A bit-string longest-common-subsequence algorithm. *Inf. Process. Lett.*, 23(6):305–310, December.

Nguyen Viet Cuong, Nan Ye, Wee Sun Lee, and Hai Leong Chieu. 2014. Conditional random field with high-order dependencies for sequence labeling and segmentation. *The Journal of Machine Learning Research*, 15(1):981–1009.

Michael Heilman and Noah A. Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 1011–1019, Los Angeles, California, USA.

Yongshuai Hou, Cong Tan, Xiaolong Wang, Yaoyun Zhang, Jun Xu, and Qingcai Chen. 2015. HITSZ-ICRC: Exploiting classification approach for answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '15, pages 196–202, Denver, Colorado, USA.

Paul Jaccard. 1901. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579.

Jiwoon Jeon, W. Bruce Croft, and Joon Ho Lee. 2005. Finding similar questions in large question and answer archives. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, pages 84–90, Bremen, Germany.

Thorsten Joachims. 1999. Making Large-scale Support Vector Machine Learning Practical. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in Kernel Methods*, pages 169–184. MIT Press, Cambridge, Massachusetts, USA.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, California, USA.

Caroline Lyon, James Malcolm, and Bob Dickerson. 2001. Detecting short passages of similar text in large document collections. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, EMNLP '01, pages 118–125, Pittsburgh, Pennsylvania, USA.

Peter McCullagh. 1980. Regression models for ordinal data. *J. Roy. Statist. Soc. B*, 42:109–142.

Alessandro Moschitti, Silvia Quarteroni, Roberto Basili, and Suresh Manandhar. 2007. Exploiting syntactic and shallow semantic kernels for question answer classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, ACL '07, pages 776–783, Prague, Czech Republic.

Alessandro Moschitti. 2006. Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Machine Learning: ECML 2006*, volume 4212 of *Lecture Notes in Computer Science*, pages 318–329. Springer Berlin Heidelberg.

Kevin Murphy. 2012. *Machine Learning A Probabilistic Perspective*. The MIT Press.

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.

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.

Filip Radlinski and Thorsten Joachims. 2005. Query chains: Learning to rank from implicit feedback. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, pages 239–248, Chicago, Illinois, USA.

Aliaksei Severyn and Alessandro Moschitti. 2012. Structural relationships for large-scale learning of answer re-ranking. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 741–750, Portland, Oregon, USA.

Aliaksei Severyn and Alessandro Moschitti. 2013. Automatic feature engineering for answer selection and extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, EMNLP '13, pages 458–467, Seattle, Washington, USA.

Aliaksei Severyn, Massimo Nicosia, and Alessandro Moschitti. 2013. Learning adaptable patterns for passage reranking. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, CoNLL '13, pages 75–83, Sofia, Bulgaria.

Dan Shen and Mirella Lapata. 2007. Using semantic roles to improve question answering. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '07, pages 12–21, Prague, Czech Republic.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2008. Learning to rank answers on large online QA collections. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics and the Human Language Technology Conference*, ACL-HLT '08, pages 719–727, Columbus, Ohio, USA.

Quan Hung Tran, Vu Tran, Tu Vu, Minh Nguyen, and Son Bao Pham. 2015. JAIST: Combining multiple features for answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '15, pages 215–219, Denver, Colorado, USA.

Mengqiu Wang and Christopher D. Manning. 2010. Probabilistic tree-edit models with structured latent variables for textual entailment and question answering. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 1164–1172, Beijing, China.

Michael Wise. 1996. Yap3: Improved detection of similarities in computer program and other texts. In *Proceedings of the Twenty-seventh SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '96, pages 130–134, New York, New York, USA.

Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013. Answer extraction as sequence tagging with tree edit distance. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '13, pages 858–867.

Chun-Nam Yu and T. Joachims. 2008. Training structural SVMs with kernels using sampled cuts. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 794–802.