Global Thread-Level Inference for Comment Classification in Community Question Answering

Shafiq Joty, Alberto Barrón-Cedeño, Giovanni Da San Martino, Simone Filice, Lluís Màrquez, Alessandro Moschitti, and **Preslav Nakov**,

Qatar Computing Research Institute, HBKU

{sjoty,albarron,gmartino,sfilice,
lmarquez,amoschitti,pnakov}@qf.org.qa

Abstract

Community question answering, a recent evolution of question answering in the Web context, allows a user to quickly consult the opinion of a number of people on a particular topic, thus taking advantage of the wisdom of the crowd. Here we try to help the user by deciding automatically which answers are good and which are bad for a given question. In particular, we focus on exploiting the output structure at the thread level in order to make more consistent global decisions. More specifically, we exploit the relations between pairs of comments at any distance in the thread, which we incorporate in a graph-cut and in an ILP frameworks. We evaluated our approach on the benchmark dataset of SemEval-2015 Task 3. Results improved over the state of the art, confirming the importance of using thread level information.

1 Introduction

Community question answering (CQA) is a recent evolution of question answering, in the Web context, where users pose questions and then receive answers from other users. This setup is very attractive, as the anonymity on the Web allows users to ask just about anything and then hope to get some honest answers from a number of people. On the negative side, there is no guarantee about the quality of the answers as people of very different background, knowledge, and with different motivation contribute answers to a given question.

Unlike traditional question answering (QA), in CQA answering takes the form of commenting in a forum. Thus, many comments are only loosely connected to the original question, and some are not answers at all, but are rather interactions between users.

As question-comment threads can get quite long, finding good answers in a thread can be time-consuming. This has triggered research in trying to automatically determine which answers might be good and which ones are likely to be bad or irrelevant. One early work going in this direction is that of Qu and Liu (2011), who tried to determine whether a question is "solved" or not, given its associated thread of comments. As a first step in the process, they performed a comment-level classification, considering four classes: *problem*, *solution*, *good feedback*, and *bad feedback*.

More recently, the shared task at SemEval 2015 on Answer Selection in CQA (Nakov et al., 2015), whose benchmark datasets we will use below, tackled the task of identifying *good*, *potential*ly useful, and *bad* comments within a thread. In that task, the top participating systems used thread-level features, in addition to the usual local features that only look at the question–answer pair. For example, the second-best team, HITSZ-ICRC, used as a feature the position of the comment in the thread (Hou et al., 2015). Similarly, our participation, which achieved the third-best postition, used features that try to describe a comment in the context of the entire comment thread, focusing on user interaction (Nicosia et al., 2015). Finally, the fifth-best team, ICRC-HIT, treated the answer selection task as a sequence labeling problem and proposed recurrent convolution neural networks to recognize good comments (Zhou et al., 2015b).

In a follow-up work, Zhou et al. (2015a) included a long-short term memory in their convolution neural network to learn the classification sequence for the thread. In parallel, in our recent work (Barrón-Cedeño et al., 2015), we tried to exploit the dependencies between the thread comments to tackle the same task. We did it by designing features that look globally at the thread and by applying structured prediction models, such as Conditional Random Fields (Lafferty et al., 2001).

Our goal in this paper goes in the same direction: we are interested in exploiting the output structure at the thread level to make more consistent global assignments.

To the best of our knowledge, there is no work in QA that 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. In particular, we focus on the relations between two comments *at any distance in the thread*. This is more general than previous approaches, which were either limited to sequential interactions or considered conversational interactions only at the level of features.

We propose a model based on the idea that similar comments should have similar labels. Below, we apply graph-cut and we compare it to an integer linear programming (ILP) formulation for decoding under global constraints; we also provide results with a linear-chain CRF. We show that the CRF is ineffective due to long-distance relations, e.g., a conversation in a thread can branch and then come back later. On the contrary, the global inference models (either graph-cut or ILP) using the similarity between pairs of comments manage to significantly improve a strong baseline performing local comment-based classifications.

2   The Task

We use the CQA-QL corpus from 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 answers for each question, i.e., *question-answer threads*. Each question, and each answer, consist of a short title and a more detailed description.   There is also meta information associated with both, e.g., ID of the user asking/answering the question, timestamp, category. The task asks participants to determine for each answer in the thread whether it is *Good*, *Bad*, or *Potential*ly useful for the given question.

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

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.

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.

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

A$_4$ Slim chance . . .

Figure 1: **Example from SemEval-2015 Task 3.**

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. In this paper, we focus on a *2-class* variant of the above Subtask A, which is closer to a real CQA application. We merge *Potential* and *Bad* labels into *Bad* and we focus on the 2-class problem: Good-vs-Bad. Table 1 shows some statistics about the resulting dataset used for development, training and testing.

Category	Train	Dev	Test
Questions	2,600	300	329
Comments	16,541	1,645	1,976
Good	8,069	875	997
Bad	8,472	770	979

Table 1: **Statistics about the CQA-QL dataset:** after merging *Bad* and *Potential* into *Bad*.

3   Our Proposed Solution

We model the pairwise relations between the comments in the answer thread ($\{c_i\}_{i=1}^{n}$) to produce a better global assignment: we combine the predictions of a Good-vs-Bad classifier at the comment level with the output of a pairwise classifier, Same-vs-Different, which takes two comments and predicts whether they should have the same label.

Each comment c_i has an individual score s_{iK}, provided by the Good-vs-Bad classifier, for being in class $K \in \{G, B\}$ (i.e., G for *Good* and B for *Bad*). Moreover, for each pair of comments (c_i, c_j), we have an association score s_{ij}, an estimate by the pairwise classifier about how likely it is that the comments c_i and c_j will have the same label. Next, we define two ways of doing global inference using these two sources of information.

3.1 Graph Partition Approach

Here our goal is to find a partition $P = (G, B)$ that *minimizes* the following cost:

$$C(P) = \lambda \Big[\sum_{c_i \in G} s_{iB} + \sum_{c_i \in B} s_{iG} \Big] + (1 - \lambda) \sum_{c_i \in G, c_j \in B} s_{ij}$$

The first part of the cost function discourages misclassification of individual comments, while the second part encourages similar comments to be in the same class. The mixing parameter $\lambda \in [0, 1]$ determines the relative strength of the two components. Our approach is inspired by Pang and Lee (2004), where they model the proximity relation between sentences for finding subjective sentences in product reviews, whereas we are interested in global inference based on local classifiers.

The optimization problem can be efficiently solved by finding a *minimum cut* of a weighted undirected graph $G = (V, E)$. The set of nodes $V = \{v_1, v_2, \cdots, v_n, s, t\}$ represent the n comments in a thread, the *source* and the *sink*. We connect each comment node v_i to the source node s by adding an edge $w(v_i, s)$ with capacity s_{iG}, and to the sink node t by adding an edge $w(v_i, t)$ with capacity s_{iB}. Finally, we add edges $w(v_i, v_j)$ with capacity s_{ij} to connect all pairs of comments.

Minimizing $C(P)$ amounts to finding a partition (S, T), where $S = \{s\} \cup S'$ and $T = \{t\} \cup T'$ for $s \notin S', t \notin T'$, that minimizes the cut capacity, i.e., the net flow crossing from S to T. One crucial advantage of this approach is that we can use *max-flow* algorithms to find the exact solution in polynomial time — near-linear in practice (Cormen et al., 2001; Boykov and Kolmogorov, 2004).

3.2 Integer Linear Programming Approach

Here we follow the *inference with classifiers* approach by Roth and Yih (2004), solved with Integer Linear Programming (ILP). We have one ILP problem per question–answer thread. We define a set of binary variables, whose assignment will univocally define the classification of all comments in the thread. In particular, we define a pair of variables for each answer: x_{iG} and x_{iB}, $1 \leq i \leq n$. Assigning 1 to x_{iG} means that comment c_i in the thread is classified as *Good*; assigning it 0 means that c_i is not classified as *Good*. The same applies to the other classes (here, only *Bad*). Also, we have a pair of variables for each pair of comments (to capture the pairwise relations): x_{ijS} and x_{ijD}, $1 \leq i < j \leq n$. Assigning 1 to x_{ijS} means that

c_i and c_j have the same label; assigning 0 to x_{ijS} means that c_i and c_j do not have the same label. The same interpretation holds for the other possible classes (in this case only *Different*).[4]

Let c_{iG} be the cost of classifying c_i as *Good*, c_{ijS} be the cost of assigning the same labels to c_i and c_j, etc. Following (Roth and Yih, 2004), these costs are obtained from local classifiers by taking log probabilities, i.e., $c_{iG} = -\log s_{iG}$, $c_{ijS} = -\log s_{ij}$, etc. The goal of the ILP problem is to find an assignment A to all variables x_{iG}, x_{iB}, x_{ijS}, x_{ijD} that *minimizes* the cost function:

$$\begin{aligned} C(A) = & \; \lambda \cdot \sum_{i=1}^{N} (c_{iG} \cdot x_{iG} + c_{iB} \cdot x_{iB}) + \\ & (1 - \lambda) \cdot \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (c_{ijS} \cdot x_{ijS} + c_{ijD} \cdot x_{ijD}) \end{aligned}$$

subject to the following constraints: (*i*) All variables are binary; (*ii*) One and only one label is assigned to each comment or pair of comments; (*iii*) The assignments to the comment variables and to the comment-pair variables are consistent: $x_{ijD} = x_{iG} \oplus x_{jG}, \forall i, j \;\; 1 \leq i < j \leq n$. $\lambda \in [0, 1]$ is a parameter used to balance the contribution of the two sources of information.

4 Local Classifiers

For classification, we use Maximum Entropy, or MaxEnt, (Murphy, 2012), as it yields a probability distribution over the class labels, which we then use directly for the graph arcs and the ILP costs.

4.1 Good-vs-Bad Classifier

Our most important features measure the similarity between the question (q) and the comment (c). We compare lemmata and POS [1-4]-grams using Jaccard (1901), containment (Lyon et al., 2001), and cosine, as well as using some similarities from DKPro (Bär et al., 2012) such as longest common substring (Allison and Dix, 1986) and greedy string tiling (Wise, 1996). We also compute similarity using partial tree kernels (Moschitti, 2006) on shallow syntactic trees.

Forty-three Boolean features express whether (*i*) c includes URLs or emails, the words "yes", "sure", "no", "neither", "okay", etc., as well as '?' and '@' or starts with "yes" (12 features); (*ii*) c includes a word longer than fifteen characters (1);

---

[4]Setting a binary variable for each class label is necessary to have an objective function that is linear on the labels.

(*iii*) q belongs to each of the forum categories (26); and (*iv*) c and q were posted by the same user (4). An extra feature captures the length of c.

Four features explore whether c is close to a comment by the user who asked the question, u_q: (*i-ii*) there is a comment by u_q following c and (not) containing an acknowledgment or (*iii*) containing a question, or (*iv*) among the comments preceding c there is one by u_q asking a question. We model dialogues by identifying conversation chains between two users with three features: whether c is at the beginning/middle/end of a chain. There are copies of these features for chains in which u_q participates. Another feature for c_{u_i} checks whether the user u_i wrote more than one comment in the current thread. Three more features fire for the first/middle/last comment by u_i. One extra feature counts the total number of comments written by u_i. Finally, there is a feature modeling the position of c in the thread.

4.2 Same-vs-Different Classifier

We use the following types of features for a pair of comments (c_i, c_j): (*i*) all the features from the Good-vs-Bad classifier (i.e., we subtracted the feature vectors representing the two comments, $|v_i - v_j|$)[5]; (*ii*) the similarity features between the two comments, $sim(c_i, c_j)$; and (*iii*) the prediction from the Good-vs-Bad classifiers on c_i and c_j (i.e., the scores for c_i and c_j, the product of the two scores, and five boolean features specifying whether any of c_i and c_j are predicted as *Good*, *Bad*, and whether their predictions are identical).

5 Experiments and Evaluation

We performed standard pre-processing, and we further filtered user's signatures. All parameters (e.g., Gaussian prior for MaxEnt and the mixing λ for the graph-cut and ILP) were tuned on the development set. We also trained a second-order linear-chain CRF to check the contribution of the sequential relations between comments. We report results on the official SemEval test set for all methods. For the Same-vs-Different problem, we explored a variant of training with three classes, by splitting the *Same* class into *Same-Good* and *Same-Bad*. At test time, the probabilities of these two subclasses are added to get the probability of *Same* and all the algorithms are run unchanged.

---

[5]Subtracting vectors is standard in preference learning (Shen and Joshi, 2003). The absolute value is necessary to emphasize comment differences instead of preferences.

Classifier	P	R	F_1	Acc
baseline: *Same*				69.26
MaxEnt-2C	73.95	90.99	81.59	71.56
MaxEnt-3C	77.15	80.42	78.75	69.94

Table 2: **Same-vs-Different classification.** P, R, and F_1 are calculated with respect to *Same*.

Table 2 shows the results for the Same-vs-Different classification. We can see that the two-class MaxEnt-2C classifier works better than the three-class MaxEnt-3C. MaxEnt-3C has more balanced P and R, but loses in both F_1 and accuracy. MaxEnt-2C is very skewed towards the majority class, but performs better due to the class imbalance. Overall, it seems very difficult to learn with the current features, and both methods only outperform the majority-class baseline by a small margin. Yet, while the overall accuracy is low, note that the graph-cut/ILP inference allows us to recover from some errors, because if nearby utterances are clustered correctly, the wrong decisions should be outvoted by correct ones.

The results for Good-vs-Bad are shown in Table 3. On the top are the best systems at SemEval-2015 Task 3. We can see that our MaxEnt classifier is competitive: it shows higher accuracy than two of them, and the highest F_1 overall.[6]

System	P	R	F_1	Acc
Top-3 at SemEval-2015 Task 3				
JAIST	80.23	77.73	78.96	79.10
HITSZ-ICRC	75.91	77.13	76.52	76.11
QCRI	74.33	83.05	78.45	76.97
Instance Classifiers				
MaxEnt	75.67	84.33	79.77	78.43
Linear Chain Classifiers				
CRF	74.89	83.45	78.94	77.53
Global Inference Classifiers				
ILP	77.04	83.53	80.15	79.14‡
Graph-cut	78.30	82.93	**80.55**	**79.80**‡
ILP-3C	78.07	80.42	79.23	78.73
Graph-cut-3C	78.26	81.32	79.76	79.19†

Table 3: **Good-vs-Bad classification.** ‡ and † mark statistically significant differences in accuracy compared to the baseline MaxEnt classifier with confidence levels of 99% and 95%, respectively (randomized test).

---

[6]This comparison is not strictly fair as the SemEval systems were trained to predict three classes, and here we remapped them to two. We just want to show that our baseline system is very strong.

The CRF model is worse than MaxEnt on all measures, which suggests that the sequential information does not help. This can be because many interactions between comments are long-distance and there are gaps in the threads due to the annotation procedure at SemEval (Nakov et al., 2015).

However, global inference with graph-cut and ILP improves both F_1 and accuracy, mostly due to better recall. Graph-cut works better than ILP as it has higher precision, which helps F_1 and accuracy.

Both yield statistically significant improvements over the MaxEnt classifier; they also improve over the state-of-the-art JAIST system. Note that the devtest-tuned values of λ for graph-cut and ILP put much lower weight to the Same-vs-Different component (values are 0.95 and 0.91, respectively). Finally, as expected, using the predictions of MaxEnt-2C in the global classifiers is better than using those from MaxEnt-3C.

Q: I have a female friend who is leaving for a teaching job in Qatar in January. What would be a useful portable gift to give her to take with her?

A$_1$ A couple of good best-selling novels. It's hard to find much here in Doha in the way of books.
Local: Good, Global: Good, Human: Good

A$_2$ ipod to entertain herself in case of boredom... a lot of patience for her students...
Local: Good, Global: Good, Human: Good

A$_3$ Thanks, please keep suggestions coming, would like to send her off with a useful gift.
Local: Bad, Global: Bad, Human: Bad

A$_6$ Bacon. Nice bread, bacon, bacon, errmmm bacon and a pork joint..
Local: Bad, Global: Good, Human: Good

A$_9$ Couple of good novels, All time favorite movies, ..
Local: Bad, Global: Good, Human: Good

A$_{11}$ Ditto on the books and dvd's. Excedrin.
Local: Bad, Global: Bad, Human: Good

A$_{12}$ Ditto on the bacon, pork sausage, pork chops, ham,..can you tell we miss pork! I think getting a care package together: her favorite perfume; shampoo; conditioner; glycerin soaps; set of DVDs of her favorite TV series.. Oh, and did I mention she should pack PATIENCE?
Local: Bad, Global: Good, Human: Good

Figure 2: **An excerpt of a thread** with decisions by local and global classifiers, and humans.

6 Discussion

We manually examined a number of examples where our global classfier could successfully recover from the errors made by the local classifier, and where it failed to do so. In Figure 2, we show the classification decisions of our local and global (graph-cut) classifiers along with the human annotations for an excerpt of a thread.

For example, consider answers A_6, A_9, and A_{12}, which were initially misclassified as *Bad* by the local classifier, but later recovered by the global classifier exploiting the *pairwise* information. In this case, the votes received by these answers from other *Good* answers in the thread for being in the *same* class won against the votes received from other *Bad* answers.

Now consider A_{11}, which our method failed to classify correctly as *Good*. Our investigation revealed that in this case the votes from the *Bad* answers won against the votes from the *Good* ones. The accuracy of the pairwise classifier has proven to be crucial for the performance of our overall framework. We probably need more informative features (e.g., textual entailment and semantic similarity to capture the relation between books and novels, movies and DVDs, etc.) in order to improve the pairwise classification performance.

7 Conclusion and Future Work

We have investigated the use of thread-level information for answer selection in CQA. We have shown that using a pairwise classifier that predicts whether two comments should get the same label, followed by a graph-cut (or ILP) global inference improves significantly over a very strong baseline as well as over the state of the art. We have further shown that using a linear-chain CRF model does not help, probably because many interactions between comments are long distance.

In future work, we would like to improve the pairwise classifiers with richer features, as this is currently the bottleneck for improving the performance in the global model. We further plan to test our framework on other CQA datasets, including on other languages.[7] Last but not least, we are interested in extending this research with even more global information, e.g., by modeling global decision consistency across multiple threads.

---

[7]SemEval-2015 Task 3 had an Arabic subtask, but there the answers were not coming from the same thread.

References

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

Daniel Bär, Chris Biemann, Iryna Gurevych, and Torsten Zesch. 2012. UKP: Computing semantic textual similarity by combining multiple content similarity measures. In *Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, pages 435–440, Montréal, Canada.

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*, ACL-IJCNLP '15, pages 687–693, Beijing, China.

Yuri Boykov and Vladimir Kolmogorov. 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(9):1124–1137, September.

Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. 2001. *Introduction to Algorithms*. McGraw-Hill Higher Education.

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, CO.

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.

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, CA.

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

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.

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, CO.

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, CO.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, pages 271–278, Barcelona, Spain.

Zhonghua Qu and Yang Liu. 2011. Finding problem solving threads in online forum. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, IJCNLP '11, pages 1413–1417, Chiang Mai, Thailand.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning*, CoNLL '04, pages 1–8, Boston, MA.

Libin Shen and Aravind K. Joshi. 2003. An SVM based voting algorithm with application to parse reranking. In *Proceedings of the Seventh Conference on Natural Language Learning*, CONLL '03, pages 9–16, Edmonton, Canada.

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, NY.

Xiaoqiang Zhou, Baotian Hu, Qingcai Chen, Buzhou Tang, and Xiaolong Wang. 2015a. Answer sequence learning with neural networks for answer selection in community question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 713–718, Beijing, China.

Xiaoqiang Zhou, Baotian Hu, Jiaxin Lin, Yang xiang, and Xiaolong Wang. 2015b. ICRC-HIT: A deep learning based comment sequence labeling system for answer selection challenge. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '15, pages 210–214, Denver, CO.