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

A1 your qid should specify what is the actual profession you have. i think for me, your chances to have a driver's license is low.

A2 dear richard, his asking if he can obtain. means he have the driver license

A3 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, and is publicly available on the task’s website. 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, where answers 2 and 4 are good, answer 1 is potentially useful, and answer 3 is bad.

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. Our first four features indicate whether $c$ appears in the proximity of a comment by $u_q$.

1http://www.qatarliving.com/forum
2http://alt.qcri.org/semeval2015/task3/

4The 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.
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.

<table>
<thead>
<tr>
<th></th>
<th>P$_m$</th>
<th>R$_m$</th>
<th>F$_{1,m}$</th>
<th>A$_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Baseline Features</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVM</td>
<td>52.96</td>
<td>53.14</td>
<td>52.87</td>
<td>67.61</td>
</tr>
<tr>
<td>OrdReg</td>
<td>53.33</td>
<td>51.54</td>
<td>51.87</td>
<td>65.38</td>
</tr>
<tr>
<td><strong>Baseline+Thread-level Features</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVM</td>
<td>56.31</td>
<td>56.46</td>
<td>56.33</td>
<td>72.27</td>
</tr>
<tr>
<td>OrdReg</td>
<td>57.68</td>
<td>57.04</td>
<td>57.20</td>
<td>72.47</td>
</tr>
<tr>
<td><strong>SemEval top three</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAILF</td>
<td>57.31</td>
<td>57.20</td>
<td>57.19</td>
<td>72.52</td>
</tr>
<tr>
<td>HITSZ</td>
<td>57.83</td>
<td>56.82</td>
<td>56.41</td>
<td>68.67</td>
</tr>
<tr>
<td>QCRI</td>
<td>54.34</td>
<td>53.57</td>
<td>53.74</td>
<td>70.50</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th></th>
<th>P$_r$</th>
<th>R$_r$</th>
<th>F$_r$</th>
<th>A$_r$</th>
<th>F$_{1,ta}$</th>
<th>A$_{ta}$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Baseline Features</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVM</td>
<td>70.58</td>
<td>84.45</td>
<td>76.89</td>
<td>74.39</td>
<td>66.52</td>
<td>76.13</td>
</tr>
<tr>
<td>SVM$_{hmm}$</td>
<td>72.57</td>
<td>85.46</td>
<td>78.49</td>
<td>76.37</td>
<td>68.55</td>
<td>77.58</td>
</tr>
<tr>
<td>LogReg</td>
<td>65.05</td>
<td>91.27</td>
<td>75.96</td>
<td>70.85</td>
<td>68.84</td>
<td>74.79</td>
</tr>
<tr>
<td>CRF$_{map}$</td>
<td>72.48</td>
<td>86.66</td>
<td>78.94</td>
<td>76.67</td>
<td>67.17</td>
<td>76.55</td>
</tr>
<tr>
<td>CRF$_{mpm}$</td>
<td>71.55</td>
<td>84.25</td>
<td>77.38</td>
<td>75.15</td>
<td>66.54</td>
<td>75.42</td>
</tr>
<tr>
<td><strong>Baseline+Thread-level Features</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVM</td>
<td>75.29</td>
<td>85.26</td>
<td>79.96</td>
<td>78.44</td>
<td>67.65</td>
<td>76.02</td>
</tr>
<tr>
<td>SVM$_{hmm}$</td>
<td>74.84</td>
<td>83.25</td>
<td>78.82</td>
<td>77.43</td>
<td>66.61</td>
<td>77.06</td>
</tr>
<tr>
<td>LogReg</td>
<td>73.32</td>
<td>80.56</td>
<td>79.39</td>
<td>77.33</td>
<td>68.10</td>
<td>75.57</td>
</tr>
<tr>
<td>CRF$_{map}$</td>
<td>77.77</td>
<td>85.76</td>
<td>79.31</td>
<td>77.43</td>
<td>66.37</td>
<td>76.08</td>
</tr>
<tr>
<td>CRF$_{mpm}$</td>
<td>74.35</td>
<td>85.46</td>
<td>79.51</td>
<td>77.78</td>
<td>67.36</td>
<td>76.63</td>
</tr>
</tbody>
</table>

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 $u_q$:

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

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 $c_{a_q}$ 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.

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.
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.

### 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$^h_{mm}$ (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 $y^* = \arg \max_{y_1:T} P(y_1:T | x_1:T)$, 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 $\hat{y} = (\arg \max_{y_1} P(y_1 | x_1:T), \ldots, \arg \max_{y_T} P(y_T | x_1:T))$. 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.

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$^h_{mm}$ 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.

---

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

<table>
<thead>
<tr>
<th>Baseline Features</th>
<th>P</th>
<th>R</th>
<th>$F_1$</th>
<th>A</th>
<th>$F_{1,ta}$</th>
<th>$A_{ta}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>68.86 ± 1.42</td>
<td>82.34 ± 1.04</td>
<td>74.98 ± 0.73</td>
<td>72.90 ± 1.00</td>
<td>64.56 ± 0.97</td>
<td>75.32 ± 0.40</td>
</tr>
<tr>
<td>SVM$^h_{mm}$</td>
<td>70.34 ± 1.57</td>
<td>81.00 ± 1.98</td>
<td>75.28 ± 1.05</td>
<td>73.75 ± 1.56</td>
<td>65.25 ± 1.16</td>
<td>74.68 ± 1.05</td>
</tr>
<tr>
<td>LogReg</td>
<td>64.20 ± 1.33</td>
<td>88.54 ± 0.81</td>
<td>74.42 ± 0.80</td>
<td>69.99 ± 0.94</td>
<td>66.00 ± 1.33</td>
<td>73.04 ± 0.96</td>
</tr>
<tr>
<td>CRF$_{map}$</td>
<td>69.11 ± 1.41</td>
<td>80.63 ± 1.76</td>
<td>74.42 ± 1.29</td>
<td>72.66 ± 1.75</td>
<td>63.90 ± 1.71</td>
<td>73.51 ± 0.73</td>
</tr>
<tr>
<td>CRF$_{mpm}$</td>
<td>69.60 ± 1.65</td>
<td>81.17 ± 1.28</td>
<td>74.93 ± 1.19</td>
<td>73.20 ± 1.77</td>
<td>64.53 ± 1.37</td>
<td>74.32 ± 0.92</td>
</tr>
</tbody>
</table>

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.).
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 Bad Doha? If so, I'd love to go.
   Expat Sueo
   P.S. Is that a labradoodle in your avatar?

Q_u_2: Good Scissor.
   Dears, anyone have an idea where to find a good scissor for hair and beard trimming please???

c_1,u_2: Visit Family food center

c_2,u_2: Al rawnaq airport road...U'll find all types of scissors there...

c_3,u_2: Thank you all ... I will try that.

Q_u_3: Visit Family food center

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,fa} (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,fa} and A_{fa}, on the 5-fold cross-validation using them is always beneficial: for F_{1,fa} 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_2), 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_2— 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


