It Takes Three to Tango: Triangulation Approach to Answer Ranking in Community Question Answering

Preslav Nakov, **Lluís Màrquez** and **Francisco Guzmán**
Arabic Language Technologies Research Group
Qatar Computing Research Institute, HBKU
{pnakov,lmarquez,fguzman}@qf.org.qa

Abstract

We address the problem of answering new questions in community forums, by selecting suitable answers to already asked questions. We approach the task as an answer ranking problem, adopting a pairwise neural network architecture that selects which of two competing answers is better. We focus on the utility of the three types of similarities occurring in the triangle formed by the original question, the related question, and an answer to the related comment, which we call *relevance, relatedness*, and *appropriateness*. Our proposed neural network models the interactions among all input components using syntactic and semantic embeddings, lexical matching, and domain-specific features. It achieves state-of-the-art results, showing that the three similarities are important and need to be modeled together. Our experiments demonstrate that all feature types are relevant, but the most important ones are the lexical similarity features, the domain-specific features, and the syntactic and semantic embeddings.

1 Introduction

In recent years, community Question Answering (cQA) forums, such as StackOverflow, Quora, Qatar Living, etc., have gained a lot of popularity as a source of knowledge and information. These forums typically organize their content in the form of multiple topic-oriented *question–comment threads*, where a question posed by a user is followed by a list of other users' comments, which intend to answer the question.

Many of such on-line forums are not moderated, which often results in (a) *noisy* and (b) *redundant* content, as users tend to deviate from the question and start asking new questions or engage in conversations, fights, etc.

Web forums try to solve problem (a) in various ways, most often by allowing users to up/down-vote answers according to their perceived usefulness, which makes it easier to retrieve useful answers in the future. Unfortunately, this negatively penalizes recent comments, which might be the most relevant and updated ones. This is due to the time it takes for a comment to accumulate votes. Moreover, voting is prone to abuse by forum trolls (Mihaylov et al., 2015; Mihaylov and Nakov, 2016a).

Problem (b) is harder to solve, as it requires that users verify that their question has not been asked before, possibly in a slightly different way. This search can be hard, especially for less experienced users as most sites only offer basic search, e.g., a site search by Google. Yet, solving problem (b) automatically is important both for site owners, as they want to prevent question duplication as much as possible, and for users, as finding an answer to their questions without posting means immediate satisfaction of their information needs.

In this paper, we address the general problem of finding good answers to a given *new question* (referred to as *original question*) in one such community-created forum. More specifically, we use a pairwise deep neural network to rank comments retrieved from different question-comment threads according to their *relevance* as answers to the original question being asked.

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1586–1597,
Austin, Texas, November 1-5, 2016.

A key feature of our approach is that we investigate the contribution of the edges in the triangle formed by the pairwise interactions between the *original question*, the *related question*, and the *related comments* to rank comments in a unified fashion. Additionally, we use three different sets of features that capture such similarity: lexical, distributed (semantics/syntax), and domain-specific knowledge.

The experimental results show that addressing the answer ranking task directly, i.e., modelling only the similarity between the original question and the answer-candidate comments, yields very low results. The other two edges of the triangle are needed to obtain good results, i.e., the similarity between the original question and the related question and the similarity between the related question and the related comments. Both aspects add significant and cumulative improvements to the overall performance. Finally, we show that the full network, including the three pairs of similarities, outperforms the state-of-the-art on a benchmark dataset.

The rest of the paper is organized as follows: Section 2 discusses the similarity triangle in answer ranking for cQA, Section 3 presents our pairwise neural network model for answering new questions in community forums, which integrates multiple levels of interaction, Section 4 describes the features we used, Section 5 presents our evaluation setup, the experiments and the results, Section 6 discusses some related work, and Section 7 wraps up the paper with a brief summary of the contributions and some possible directions for future work.

2 The Similarity Triangle in cQA

Figure 1 presents an example illustrating the similarity triangle that we use when solving the answer ranking problem in cQA. In the figure, q stands for the new question, q' is an existing related question, and c is a comment within the thread of question q'.

The edge $\overline{qc}$ relates to the main cQA task addressed in this paper, i.e., deciding whether a comment for a potentially related question is a good answer to the original question. We will say that the relation captures the *relevance* of c for q.

The edge $\overline{qq'}$ represents the similarity between the original and the related questions. We will call this relation *relatedness*.

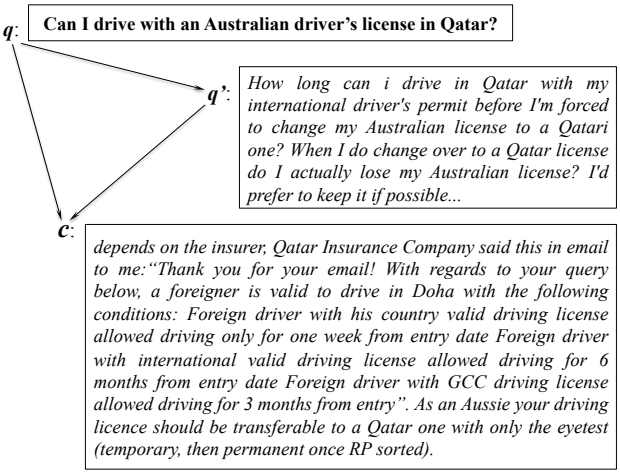

Figure 1: The similarity triangle in cQA.

Finally, the edge $\overline{q'c}$ represents the decision of whether c is a good answer for the question from its thread, q'. We will call this relation *appropriateness*.

In this particular example, q and q' are indeed related, and c is a good answer for both q' and q.[1]

In the past, the approaches to cQA were focused on using information from the new question q, an existing related question q', and a comment c within the thread of q', to solve different cQA sub-tasks. For example, *answer selection*, which selects the most appropriate comment c within the thread q', was addressed in SemEval-2015 Task 3 (Nakov et al., 2015). Similarly, *question–question similarity*, which looks for the most related questions to a given question, was addressed by many authors (Jeon et al., 2005; Duan et al., 2008; Li and Manandhar, 2011; Zhou et al., 2015; dos Santos et al., 2015).

In this paper, we solve the cQA task problem[2] in a novel way by using the three types of similarities jointly. Our main hypothesis is that *relevance*, *appropriateness*, and *relatedness* are essential to finding the best answer in a community Question Answering setting. Below we present experimental results that support this hypothesis.

[1]The essence of this triangle is also described in SemEval 2016 Task 3 to motivate a three-subtask setting for cQA (Nakov et al., 2016). In that evaluation exercise, $\overline{q'c}$ and $\overline{qq'}$ are presented as subtask A and subtask B, respectively. In this paper, we mainly use them as similarity relations to be modeled in the learning architecture to solve the answer ranking task.

[2]We use the task setup and the datasets from SemEval-2016 Task 3, focusing on subtask C (Nakov et al., 2016).

3 Neural Model for Answer Ranking

As explained above, we tackle answer ranking as a three-way similarity problem, exploring similarity features that capture lexical, distributed (semantics and syntax), and domain-specific knowledge. To achieve this, we propose a pairwise neural network (NN) approach for the cQA task, which is inspired by our NN framework for machine translation evaluation (Guzmán et al., 2015).[3] The input of the NN consists of the original question q, two competing comments, c_1 and c_2, and the questions from the threads of the two comments, q'_1 and q'_2. The output of the network is a decision about which of the two comments is a better answer to q.

The main properties of our NN approach can be summarized as follows: (*i*) it works in a pairwise fashion, which is appropriate for the ranking nature of the cQA problem; (*ii*) it allows for an easy incorporation of rich syntactic and semantic embedded representations of the input texts; (*iii*) it models non-linear relationships between all input elements (q, c_1, c_2, q'_1 and q'_2), which allows us to study the interactions and the impact of the three types of similarity (relevance, relatedness and appropriateness) when solving the answer ranking task.

3.1 Architecture

Our full NN model for pairwise answer ranking is depicted in Figure 2. We have a binary classification task with input $x = (q, q'_1, c_1, q'_2, c_2)$, which should output 1 if c_1 is a better answer to the original question q than c_2, and 0 otherwise.[4] In this setting, q'_1 and q'_2 are questions related to q, whose threads contain the comments c_1 and c_2, respectively. They provide useful information to link the two comments to the original question. On the one hand, they allow to predict whether the comments are good answers within their respective threads. On the other hand, they allow to infer whether the questions for which the comments were produced are closely related to the original question. The pair of comments can belong to the same thread (i.e., $q'_1 \equiv q'_2$) or they can come from different threads.

[3] Also, we previously used a similar framework for finding good answers in a question-comment thread (Guzmán et al., 2016a; Guzmán et al., 2016b).

[4] In this work, we do not learn to predict ties, and ties are excluded from our training data.

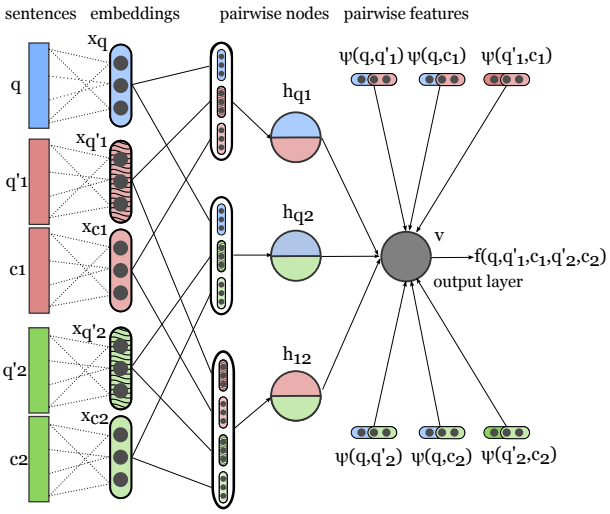

Figure 2: The overall architecture of our neural network model for pairwise answer ranking in community question answering.

The feed-forward neural network computes a sigmoid function $f(q, q'_1, c_1, q'_2, c_2) = \text{sig}(\mathbf{w_v^T}\phi(q, q'_1, c_1, q'_2, c_2) + b_v)$, where $\phi(.)$ transforms the input through the hidden layer, $\mathbf{w_v}$ are the weights from the hidden layer to the output layer, and b_v is a bias term. The function $\phi(.)$ is actually a concatenation of three subfunctions: $\phi(q, q'_1, c_1, q'_2, c_2) = [\phi_1(q, q'_1, c_1), \phi_2(q, q'_2, c_2), \phi_{1,2}(q'_1, c_1, q'_2, c_2)]$.

We first map the question and the comments to a fixed-length vector $[\mathbf{x}_q, \mathbf{x}_{q'_1}, \mathbf{x}_{c_1}, \mathbf{x}_{q'_2}, \mathbf{x}_{c_2}]$ using syntactic and semantic embeddings. Then, we feed this vector as input to the neural network, which models several types of interactions, using different groups of nodes in the hidden layer. Overall, we make use of three different groups of nodes in the hidden layer.

The first two groups include the *relevance* nodes h_{q1} and h_{q2}. These groups of hidden nodes model how relevant comment c_j is to the original question q given that it belongs to the thread of the related question q'_j. In these hidden nodes, we model complex non-linear interactions between the distributed representations of q, q'_j and c_j. Intuitively, these nodes are designed to learn to distinguish a relevant comment by extracting features from the distributed representations of a comment and of the question it is supposed to answer.

The last group of nodes in the hidden layer is the *similarity* node h_{12}. It measures the similarity between c_1 and c_2 and their respective questions q'_1 and q'_2. This node is designed to compute the non-linear interactions between the syntactic and semantic representations of comment-comment, comment-question and question-question pairs. Intuitively, this can help disambiguate when comments are very similar or were generated from the same or from very similar questions.

The model further allows to incorporate external sources of information in the form of *skip arcs* that go directly from the input to the output layer, skipping the hidden layer. These arcs represent pairwise *similarity* feature vectors inspired by the edges of the triangle in Figure 1. In Figure 2, we indicate these pairwise external feature sets as: $\psi(q, q'_1)$, $\psi(q, q'_2)$ for *relatedness*; $\psi(q'_1, c_1)$, $\psi(q'_2, c_2)$ for *appropriateness*; and $\psi(q, c_1)$, $\psi(q, c_2)$ for *relevance*. When including the *skip-arc* features, the activation at the output is $f(q, q'_1, c_1, q'_2, c_2) = \text{sig}(\mathbf{w_v^T}[\phi(q, q'_1, c_1, q'_2, c_2), \psi(q, q'_1), \psi(q, q'_2), \psi(q'_1, c_1), \psi(q'_2, c_2), \psi(q, c_1), \psi(q, c_2)] + b_v)$.

We use these feature vectors to encode machine translation evaluation measures, components thereof, cQA task-specific features, etc. The next section gives more detail about these features.

4 Features

We experiment with three kinds of features: (*i*) lexical features that measure similarity at a word, word n-gram, and paraphrase level, (*ii*) distributed representations that measure similarity at a syntactic and semantic level, (*iii*) domain-specific knowledge features, which capture similarity using thread-level information and other features that have proven valuable to solve similar tasks (Nicosia et al., 2015).

4.1 Lexical similarity features

These types of features measure similarity at a surface level between the following pairs: (q,q'_1), (q,q'_2), (q'_1, c_1), (q'_2, c_2), (q_1, c_1), and (q_2, c_2). They are inspired by our previous work on Machine Translation Evaluation (MTE) (Guzmán et al., 2015), and we previously found them useful for finding good answers in a question-comment thread (Guzmán et al., 2016a; Guzmán et al., 2016b).

MTFEATS We use (as pairwise features) the following six machine translation evaluation features: (*i*) BLEU: This is the most commonly used measure for machine translation evaluation, which is based on n-gram overlap and length ratios (Papineni et al., 2002). (*ii*) NIST: This measure is similar to BLEU, and is used at evaluation campaigns run by NIST (Doddington, 2002). (*iii*) TER: Translation error rate; it is based on the edit distance between a translation hypothesis and the reference (Snover et al., 2006). (*iv*) METEOR: A complex measure, which matches the hypothesis and the reference using synonyms and paraphrases (Lavie and Denkowski, 2009). (*v*) Unigram PRECISION and RECALL.

BLEUCOMP Following (Guzmán et al., 2015), we further use as features various components that are involved in the computation of BLEU: n-gram precisions, n-gram matches, total number of n-grams (n=1,2,3,4), lengths of the hypotheses and of the reference, length ratio between them, and BLEU's brevity penalty. Again, these are computed over the same six pairs of vectors as before.

4.2 Distributed representations

We use the following vector-based embeddings of all input components: q, c_1, c_2, q'_1, and q'_2.

GOOGLE_VEC We use the pre-trained, 300-dimensional embedding vectors from WORD2VEC (Mikolov et al., 2013). We compute a vector representation of the text by simply averaging over the embeddings of all words in the text.

QL_VEC We train in-domain word embeddings using WORD2VEC on all available QatarLiving data. Again, we use these embeddings to compute 100-dimensional vector representations for all input components by averaging over all words in the texts.

SYNTAX_VEC We parse the entire question/comment using the Stanford neural parser (Socher et al., 2013), and we use the final 25-dimensional vector that is produced internally as a by-product of parsing.

Moreover, we use the above vectors to calculate pairwise similarity features, i.e., the cosine between the following six vector pairs: (q, c_1), (q, c_2), (q'_1, c_1), (q'_2, c_2), (q, q'_1) and (q, q'_2).

4.3 Domain-specific features

We extract various domain-specific features that use thread-level and other useful information known to capture *relatedness* and *appropriateness*.

SAME_AUTHOR We have a thread-level meta-feature, which we apply to the pairs (q'_1, c_1), (q'_2, c_2). It checks whether the person answering the question is also the one who asked it, i.e., do the related question and the comment have the same author. The idea is that the person asking a question is unlikely to answer his/her own question, but s/he could ask a clarification question or thank another person who has provided a useful answer earlier in the thread.

CQ′RANK_FEAT We further have two thread-level meta-features related to the rank of the comment in the thread, which we apply to the pairs (q'_1, c_1) and (q'_2, c_2): (*i*) reciprocal rank of the comment in the thread, i.e., $1/\rho$, where ρ is the rank of the comment; (*ii*) percentile of the number of comments in the thread, calculated as follows: the first comment gets the score of 1.0, the second one gets 0.9, and so on. Note that in our dataset, there are exactly ten comments per thread.

QQ′RANK_FEAT We also have three features modeling the rank of the related question in the list of related questions for the original question, which we apply to the pairs (q, q'_1) and (q, q'_2).

In total, use the following six features: (*i*) the reciprocal rank of q'_1 or q'_2 in the list of related questions for q; (*ii*) the reciprocal ordinal rank[5] of q'_1 or q'_2 in the list of related questions for q; (*iii*) the percentile of the q'_1 or q'_2 in the list of related questions for q, calculated as for the comments.

CQRANK_FEAT. Finally, we have features for the rank of the comment in the list of 100 comments for the original question, which we apply to the pairs (q, c_1) and (q, c_2): (*i*) reciprocal rank of the comment in the list; (*ii*) percentile of the comment in the list.

TASK_FEAT. We further have features that have been proven useful in the answer selection task from SemEval 2015 Task 3 (Nakov et al., 2015). This includes some comment-specific features, which refer to c_1 and c_2 only, but which we apply twice, to generate features for the pairs (q'_1, c_1), (q'_2, c_2), (q_1, c_1), and (q_2, c_2): number of URLs/images/emails/phone numbers; number of occurrences of the string *thank*;[6] number of tokens/sentences; average number of tokens; number of nouns/verbs/adjectives/adverbs/pronouns; number of positive/negative smileys; number of single/double/triple exclamation/ interrogation symbols; number of interrogative sentences (based on parsing); number of words that are not in word2vec's Google News vocabulary.[7]

And also some question-comment pair features, which we apply to the pairs (q'_1, c_1), (q'_2, c_2), (q_1, c_1), and (q_2, c_2): (*i*) question to comment count ratio in terms of sentences/tokens/nouns/verbs/adjectives/adverbs/pronouns; (*ii*) question to comment count ratio of words that are not in word2vec's Google News vocabulary.

5 Experiments and Results

We experimented with the data from SemEval-2016 Task 3 on "Community Question Answering". More precisely, the problem addressed is subtask C (*Question–External Comment Similarity*), which is the primary cQA task. For a given new question (referred to as the *original question*), the task provides the set of the first ten related questions (retrieved by a search engine), each associated with the first ten comments appearing in the question-comment thread. The goal then is to rank the total of 100 comments according to their appropriateness with respect to the original question.

In this framework, the retrieval part of the task is done as a pre-processing step, and the challenge is to learn to rank all *good* comments above all *bad* ones. All the data comes from the QatarLiving forum, and the related questions are obtained using Google search with the original question's text limited to the `www.qatarliving.com` domain.

[5]The related questions are obtained using a query to a search engine (using words from the original question), with results limited to QatarLiving. However, some of the returned results pointed to the wrong (non-forum) sections of the website or to questions with less than ten comments, and these were skipped. Suppose that the surviving top ten related questions were at ranks 3, 7, 18, ... in the original list. Now, we can use these ranks ρ, or we can use instead the ordinal ranks r: 1, 2, 3, ...

[6]When an author thanks somebody, this post is typically a bad answer to the original question.

[7]Can detect slang, foreign language, etc., which would indicate a bad answer.

The task offers a higher quality training dataset TRAIN-PART1, which includes 200 original questions, 1,999 related questions and 19,990 comments, and a lower-quality TRAIN-PART2, which we did not use. Additionally, it provides a development set (DEV, with 50 original questions, 500 related questions and 5,000 related comments) and a TEST set (70 original questions, 700 related questions and 7,000 related comments). Apart from the class labels for subtask C, the datasets also offer class labels for subtask A (i.e., whether a comment is a good answer to the question in the thread) and subtask B (i.e., whether the related questions is relevant for the original question).

5.1 Setting

we use Theano (Bergstra et al., 2010) to train our model on TRAIN-PART1 with hidden layers of size 3 for 100 epochs with minibatches of size 30, regularization of 0.05, and a learning rate of 0.01, using stochastic gradient descent with adagrad (Duchi et al., 2011). We normalize the input feature values to the $[-1; 1]$ interval using minmax, and we initialize the NN weights by sampling from a uniform distribution as in (Bengio and Glorot, 2010).

We evaluate the model on DEV after each epoch, and ultimately we keep the model that achieves the highest accuracy;[8] in case of a tie, we prefer the parameters from a later epoch. We selected the above parameter values on the DEV dataset using the full model, and we use them for all experiments in Section 5.3, where we evaluate on the TEST dataset.

Note that, we train the NN using all pairs of (Good, Bad) comments, in both orders, ignoring ties. At test time, we compute the full ranking of comments by scoring all possible pairs, and by then accumulating the scores at the comment level.

5.2 Evaluation and baselines

The results are calculated with the official scorer from the SemEval-2016 Task 3. We report three ranking-based measures that are commonly accepted in the IR community: Mean average precision (MAP), which is the official evaluation measure of the task, average recall (AvgRec), and mean reciprocal rank (MRR).

[8] We tried Kendall's Tau (τ), but it performed slightly worse.

For comparison purposes, we report the results for two baselines. One corresponds to a random ordering of the comments, assuming zero knowledge of the task. The second one is a more realistic baseline, which keeps the question ranking from the search engine (Google search) and the chronological order of the comments within the thread of teh related question. Although this may be considered a very naïve baseline, it is actually notably informed. The question ranking from Google search takes into account the relevance of the entire thread (question and comments) to the original question. Moreover, there is a natural concentration of the best answers in the first comments of the threads.

5.3 Main results

Table 1 shows the evaluation results on the TEST dataset for several variants of our pairwise neural network architecture. Regarding our network configurations, we present the results from simpler to more complex.

Relevance The "Relevance only" network contains only the *relevance* relations and features corresponding to q, c_1 and c_2. The rest of the components are deactivated in the network. This corresponds to solving the task without any information about the related questions and the appropriateness of the comments in their threads, i.e., just by comparing the texts of the comments and of the original question. In some sense, this setup is largely less informed than the IR baseline. The results are very low, being only $\sim$7 MAP points higher than the random baseline.

Relevance + appropriateness Adding the *appropriateness* interactions between c_1 and q'_1, and between c_2 and q'_2 improves MAP by $\sim$9 points. Although more informed, as some information from the related questions is taken indirectly, the results of this system are still below the IR baseline.

Relevance + relatedness Adding the *relatedness* interactions and features between q and q'_1, and q and q'_2, turns out to be crucial. When added to the "Relevance only" basic system, the MAP score jumps to 52.43, significantly above the IR baseline. This shows that *question–question* similarity plays an important role in solving the cQA task.

System	MAP	AvgRec	MRR
Relevance relations only	21.78	20.66	22.59
+ Appropriateness	30.94	29.86	35.02
+ Relatedness	52.43	57.05	60.14
Full Network	**54.51**	**60.93**	**62.94**
Baseline 1 (random)	15.01	11.44	15.19
Baseline 2 (IR+chron.)	40.36	45.97	45.83

Table 1: Results on the answer ranking task of our full NN vs. variants using partial information.

Full Network Adding both *appropriateness* and *relatedness* interactions yields an improvement of another two MAP points absolute (to 54.51), which shows that *appropriateness* features encode information that is complementary to the information modeled by *relevance* and *relatedness*. Note that the results with the other evaluation metrics (AvgRec and MRR) follow exactly the same pattern. In summary, we can conclude that in order to solve the community question answering problem, we need to (*i*) find the best related questions, and (*ii*) judge the relevance of individual comments with respect to the new question.

5.4 Features in perspective

Table 2 shows the results of an ablation study when removing some groups of features.[9] More specifically, we drop lexical similarities, domain-specific features, and the complex semantic-syntactic interactions modeled in the hidden layer between the embeddings and the domain-specific features.

We can see that the lexical similarity features (which we modeled by MT evaluation metrics), have a large impact: excluding them from the network yields a decrease of over eight MAP points. This can be explained as the strong dependence that *relatedness* has over strict word matching. Since questions are relatively short, a better related question will be one that matches better the original question.

[9]Note that here we only show the impact of *groups* of features, e.g., we do not consider experiments with different embeddings such as GOOGLE_VEC, QL_VEC, and SYNTAX_VEC, which all belong to the lexical similarity group of features. This is because in previous work (which was limited to subtask A), our ablation study has shown that all features in a group clearly contribute to the overall performance (Guzmán et al., 2016a; Guzmán et al., 2016b).

System	MAP	AvgRec	MRR	Δ_{MAP}
Full Network	**54.51**	**60.93**	**62.94**	
– Lexical similarity	45.89	51.54	53.29	-8.62
– Domain-specific	48.48	50.46	53.78	-6.03
– Distributed rep.	51.17	56.63	56.91	-3.34
No hidden layer	52.19	58.23	59.95	-2.32

Table 2: Results of the ablation study.

As expected, eliminating the domain-specific features also hurts the performance greatly: by six MAP points absolute. Eliminating the use of distributed representation has a lesser impact: 3.3 MAP points absolute. This is in line with our previous findings (Guzmán et al., 2015; Guzmán et al., 2016a; Guzmán et al., 2016b) that semantic and syntactic embeddings are useful to make a fine-grained distinction between comments (*relevance*, *appropriateness*), which are usually longer.

We have also found that there is an interaction between features and similarity relations. For example, for *relatedness*, lexical similarity is 2.6 MAP points more informative[10] than distributed representations. In contrast, for *relevance*, distributed representations are 0.7 MAP points more informative than lexical similarities.

5.5 Impact of the hidden layer

Table 2 also presents the results of a system that has the full set of features, but eliminates the hidden layer from the neural network. This is equivalent to training a Maximum Entropy classifier with the complete set of features. This simplified system performs consistently worse than the full NN model (-2.32 MAP, -2.7 AvgRec, and -2.99 MRR points), which shows that using the hidden layer to model the non-linear interactions between information sources has a decent overall contribution.

5.6 Making appropriateness more useful

Since the SemEval-2016 Task 3 datasets also provide labeled examples for the so called "subtask A" ($\overline{q'c}$; appropriateness) and "subtask B" ($\overline{qq'}$; relatedness), one could use this supervision to help train the neural network for the primary cQA task. We observed that *relatedness* has proven quite informative. However, the improvements observed from using *appropriateness* were more modest.

[10]As measured by the relative drop in MAP performance.

System	MAP	AvgRec	MRR
Full Network	54.51	60.93	62.94
Full + *appr.* preds.	**55.82**	**61.63**	**62.39**

Table 3: Using *appropriateness* predictions.

We present here a stacked experiment in which an additional neural network trained to predict *appropriateness* is used to inform the full network model. More concretely, we train a feed-forward pairwise neural network for subtask A, which is a simplification of the architecture from Figure 2. The input is reduced to three elements (q', c_1, c_2), where q' is the thread question and c_1 and c_2 are a pair of comments in the thread. The output consists of deciding whether c_1 is a better answer to q' than c_2. All the pairwise interactions between input components are included in the hidden layer, and we use the same features to train the network as the ones described in Section 4 (obviously, this time the input and the features are reduced to those involving q', c_1 and c_2). We used this exact setting in previous work for solving subtask A (Guzmán et al., 2016a; Guzmán et al., 2016b).

We used the network to classify all subtask A examples in TRAIN-PART1, DEV and TEST, and we used the resulting scores at the comment level as skip-arc features for the full NN model: (a) alone, included in $\psi(q'_1, c_1)$ and $\psi(q'_2, c_2)$, and (b) multiplied by each of the QQ′Rank_feat features, included in $\psi(q, c_1)$ and $\psi(q, c_2)$.

In Table 3, we observe that using the pre-trained network to incorporate subtask A predictions as features yields another sizable improvement to a final MAP of 55.82 (the increase is smaller for AvgRec, and MRR is slightly hurt), which suggests that pre-training parts of the NN with labeled examples to perform a dedicated task, is a promising direction for future work.

5.7 Results in perspective

Next, in order to put our results in perspective, we compare them to the state of the art for this problem, represented by the systems that participated in SemEval-2016 Task 3, subtask C. The comparison is shown in Table 4, where we list the top-3 systems, as well as the average and the worst scores for the official runs of all participating teams.

System	MAP	AvgRec	MRR
Full Network + subtask A preds.	**55.82**	**61.63**	62.39
* 1st (Mihaylova et al., 2016)	55.41	60.66	61.48
Full Network	54.51	60.93	**62.94**
* 2nd (Filice et al., 2016)	52.95	59.27	59.23
* 3rd (Mihaylov and Nakov, 2016b)	51.68	53.43	55.96
...	...	...	...
SemEval Average	49.30	53.74	54.39
...	...	...	...
SemEval Worst	43.20	47.96	47.79
Baseline 2 (IR+chron.)	40.36	45.97	45.83

Table 4: Comparative results with the state of the art, i.e., the top-3 systems that participated in SemEval-2016 Task 3, subtask C.

We can see that all systems in the competition performed over the IR baseline with MAP scores ranging from 43.20 to 55.41. We can further see that our full network with subtask A predictions achieves the best results with 55.82 MAP. The margin over the best SemEval system is small in terms of MAP but more noticeable in terms of AvgRec and MRR. Note that, even without the Subtask A predictions, our pairwise neural network still produces results that are on par with the state of the art (with improvements slightly over one point in both cases).

6 Related Work

Recently, a variety of neural network models have been applied to community question answering tasks such as *question-question similarity* (Zhou et al., 2015; dos Santos et al., 2015; Lei et al., 2015) and *answer selection* (Severyn and Moschitti, 2015; Wang and Nyberg, 2015; Feng et al., 2015; Tan et al., 2015; Filice et al., 2016; Barrón-Cedeño et al., 2016; Mohtarami et al., 2016). Most of these papers concentrate on constructing advanced neural network architectures in order to model the problem at hand better.

For instance, dos Santos et al. (2015) propose a neural network approach combining a convolutional neural network and a bag-of-words representation for modeling question-question similarity. Similarly, Tan et al. (2015) adopt a neural attention mechanism over bidirectional long short-term memory (LSTM) neural network to generate better answer representations given the questions.

Similarly, Lei et al. (2015) use a combination of recurrent and convolutional neural models to map questions to semantic representations. The models are pre-trained within an encoder-decoder framework (from body to title) in order to de-noise the long question body from irrelevant text.

The main objective of our work here is different: we focus on studying the impact of the different input components in a novel cQA setting of ranking answers for new questions, and we use a more standard neural network.

The setting of cQA as a triangle of three interrelated subtasks, which we use here, has been recently proposed in SemEval-2016 Task 3 on *Community Question Answering* (Nakov et al., 2016). Above, we empirically compared our results to those of the best participating systems. Unfortunately, most of the systems that took part in the competition, including the winning system of the SUper team (Mihaylova et al., 2016), approached the task indirectly by solving subtask A at the thread level and then using these predictions together with the reciprocal rank of the related questions to produce a final ranking for subtask C.

One exception is the *Kelp* system (Filice et al., 2016), which was ranked second in the competition. Their approach is most similar to ours, as it also tries to combine information from different subtasks and from all input components. It does so in a modular kernel function, including stacking from independent subtask A and B classifiers, and it applies SVMs to train a Good vs. Bad classifier (Filice et al., 2016). In contrast, our approach here proceeds in a pairwise setting, it is lighter in terms of features engineering, and presents a direct way to combine the relations between the different subtasks in an integrated neural network model.

Finally, our model uses lexical features derived from machine translation evaluation. Some previous work also used MT model(s) as a feature(s) (Berger et al., 2000; Echihabi and Marcu, 2003; Jeon et al., 2005; Soricut and Brill, 2006; Riezler et al., 2007; Li and Manandhar, 2011; Surdeanu et al., 2011; Tran et al., 2015; Hoogeveen et al., 2016; Wu and Zhang, 2016), e.g., a variation of IBM model 1 (Brown et al., 1993), to compute the probability that the question is a "translation" of the candidate answer.

7 Conclusion

We presented a neural-based approach to a novel problem in cQA, where given a new question, the task is to rank comments from related question-threads according to their relevance as answers to the original question. We explored the utility of three types of similarities between the original question, the related question, and the related comment.

We adopted a pairwise feed-forward neural network architecture, which takes as input the original question and two comments together with their corresponding related questions. This allowed us to study the impact and the interaction effects of the question-question *relatedness* and comment-to-related question *appropriateness* relations when solving the primary cQA *relevance* task. The large performance gains obtained from using *relatedness* features show that question-question similarity plays a crucial role in finding relevant comments (+30 MAP points). Yet, including *appropriateness* relations is needed to achieve state-of-the-art results (+3.3 MAP) on benchmark datasets.

We also studied the impact of several types of features, especially domain-specific features, but also lexical features and syntactic embeddings. We observed that lexical similarity MTE features prove the most important, followed by domain-specific features, and syntactic and semantic embeddings. Overall, they all showed to be necessary to achieve state-of-the-art results.

In future work, we plan to use the labels for subtasks A and B, which are provided in the datasets in order to pre-train the corresponding components of the full network for answer ranking. We further want to apply a similar network to other semantic similarity problems, such as textual entailment.

Acknowledgments

This research was performed by the Arabic Language Technologies (ALT) group at the Qatar Computing Research Institute (QCRI), HBKU, part of Qatar Foundation. It is part of the Interactive sYstems for Answer Search (Iyas) project, which is developed in collaboration with MIT-CSAIL.

Last but not least, we would also like to thank the anonymous reviewers for their constructive comments, which have helped us improve the paper.

References

Alberto Barrón-Cedeño, Giovanni Da San Martino, Shafiq Joty, Alessandro Moschitti, Fahad A. 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, CA.

Yoshua Bengio and Xavier Glorot. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of Artificial Intelligence and Statistics*, AISTATS '10, pages 249–256, Chia Laguna Resort, Sardinia, Italy.

Adam Berger, Rich Caruana, David Cohn, Dayne Freitag, and Vibhu Mittal. 2000. Bridging the lexical chasm: Statistical approaches to answer-finding. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, pages 192–199, Athens, Greece.

James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. 2010. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference*, SciPy '10, Austin, TX.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Comput. Linguist.*, 19(2):263–311.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, pages 138–145, San Diego, CA.

Cicero dos Santos, Luciano Barbosa, Dasha Bogdanova, and Bianca Zadrozny. 2015. Learning hybrid representations to retrieve semantically equivalent questions. 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 694–699, Beijing, China.

Huizhong Duan, Yunbo Cao, Chin-Yew Lin, and Yong Yu. 2008. Searching questions by identifying question topic and question focus. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, ACL '08, pages 156–164, Columbus, OH.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159.

Abdessamad Echihabi and Daniel Marcu. 2003. A noisy-channel approach to question answering. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, ACL '03, pages 16–23, Sapporo, Japan.

Minwei Feng, Bing Xiang, Michael R. Glass, Lidan Wang, and Bowen Zhou. 2015. Applying deep learning to answer selection: A study and an open task. In *Proceedings of the 2015 IEEE Workshop on Automatic Speech Recognition and Understanding*, ASRU '15, pages 813–820, Scottsdale, AZ.

Simone Filice, Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2016. KeLP at SemEval-2016 Task 3: Learning semantic relations between questions and answers. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, pages 1116–1123, San Diego, CA.

Francisco Guzmán, Shafiq Joty, Lluís Màrquez, and Preslav Nakov. 2015. Pairwise neural machine translation evaluation. 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 805–814, Beijing, China.

Francisco Guzmán, Lluís Màrquez, and Preslav Nakov. 2016a. Machine translation evaluation meets community question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, ACL '16, pages 460–466, Berlin, Germany.

Francisco Guzmán, Preslav Nakov, and Lluís Màrquez. 2016b. MTE-NN at SemEval-2016 Task 3: Can machine translation evaluation help community question answering? In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, pages 887–895, San Diego, CA.

Doris Hoogeveen, Yitong Li, Huizhi Liang, Bahar Salehi, Timothy Baldwin, and Long Duong. 2016. UniMelb at SemEval-2016 Task 3: Identifying similar questions by combining a CNN with string similarity measures. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, pages 851–856, San Diego, CA.

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.

Alon Lavie and Michael Denkowski. 2009. The METEOR metric for automatic evaluation of machine translation. *Machine Translation*, 23(2–3):105–115.

Tao Lei, Hrishikesh Joshi, Regina Barzilay, Tommi S. Jaakkola, Kateryna Tymoshenko, Alessandro Mos-

chitti, and Lluís Màrquez. 2015. Denoising bodies to titles: Retrieving similar questions with recurrent convolutional models. *arXiv preprint arXiv:1512.05726*.

Shuguang Li and Suresh Manandhar. 2011. Improving question recommendation by exploiting information need. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, ACL '11, pages 1425–1434, Portland, OR.

Todor Mihaylov and Preslav Nakov. 2016a. Hunting for troll comments in news community forums. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, ACL '16, pages 399–405, Berlin, Germany.

Todor Mihaylov and Preslav Nakov. 2016b. SemanticZ at SemEval-2016 Task 3: Ranking relevant answers in community question answering using semantic similarity based on fine-tuned word embeddings. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, pages 879–886, San Diego, CA.

Todor Mihaylov, Georgi Georgiev, and Preslav Nakov. 2015. Finding opinion manipulation trolls in news community forums. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, CoNLL '15, pages 310–314, Beijing, China.

Tsvetomila Mihaylova, Pepa Gencheva, Martin Boyanov, Ivana Yovcheva, Todor Mihaylov, Momchil Hardalov, Yasen Kiprov, Daniel Balchev, Ivan Koychev, Preslav Nakov, Ivelina Nikolova, and Galia Angelova. 2016. SUper Team at SemEval-2016 Task 3: Building a feature-rich system for community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, pages 836–843, San Diego, CA.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '13, pages 746–751, Atlanta, GA.

Mitra Mohtarami, Yonatan Belinkov, Wei-Ning Hsu, Yu Zhang, Tao Lei, Kfir Bar, Scott Cyphers, and Jim Glass. 2016. SLS at SemEval-2016 Task 3: Neural-based approaches for ranking in community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, pages 828–835, San Diego, CA.

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.

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, CA.

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 '2015, pages 203–209, Denver, CO.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, ACL '02, pages 311–318, Philadelphia, PA.

Stefan Riezler, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu Mittal, and Yi Liu. 2007. Statistical machine translation for query expansion in answer retrieval. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, ACL '07, pages 464–471, Prague, Czech Republic.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 373–382, Santiago, Chile.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Biennial Conference of the Association for Machine Translation in the Americas*, AMTA '06, pages 223–231, Cambridge, MA.

Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. 2013. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, ACL '13, pages 455–465, Sofia, Bulgaria.

Radu Soricut and Eric Brill. 2006. Automatic question answering using the web: Beyond the factoid. *Inf. Retr.*, 9(2):191–206.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2011. Learning to rank answers to non-factoid questions from web collections. *Comput. Linguist.*, 37(2):351–383.

Ming Tan, Bing Xiang, and Bowen Zhou. 2015. LSTM-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108*.

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, CO.

Di Wang and Eric Nyberg. 2015. A long short-term memory model for answer sentence selection in 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 707–712, Beijing, China.

Yunfang Wu and Minghua Zhang. 2016. ICL00 at SemEval-2016 Task 3: Translation-based method for CQA system. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, pages 857–860, San Diego, CA.

Guangyou Zhou, Tingting He, Jun Zhao, and Po Hu. 2015. Learning continuous word embedding with metadata for question retrieval 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 250–259, Beijing, China.