Machine Translation Evaluation Meets Community Question Answering

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

Abstract

We explore the applicability of machine translation evaluation (MTE) methods to a very different problem: answer ranking in community Question Answering. In particular, we adopt a pairwise neural network (NN) architecture, which incorporates MTE features, as well as rich syntactic and semantic embeddings, and which efficiently models complex non-linear interactions. The evaluation results show state-of-the-art performance, with sizeable contribution from both the MTE features and from the pairwise NN architecture.

1 Introduction and Motivation

In a community Question Answering (cQA) task, we are given a question from a community forum and a thread of associated text comments intended to answer the given question; and the goal is to rank the comments according to their appropriateness to the question. Since cQA forum threads are noisy (e.g., because over time people tend to engage in discussion and to deviate from the original question), as many comments are not answers to the question, the challenge lies in learning to rank all *good* comments above all *bad* ones.

Here, we adopt the definition and the datasets from SemEval–2016 Task 3 (Nakov et al., 2016) on "Community Question Answering", focusing on subtask A (Question-Comment Similarity) only.[1] See the task description paper and the task website[2] for more detail. An annotated example is shown in Figure 1.

In this paper, we tackle the task from a novel perspective: by using ideas from machine translation evaluation (MTE) to decide on the *quality* of a comment. In particular, we extend our MTE neural network framework from (Guzmán et al., 2015), showing that it is applicable to the cQA task as well. We believe that this neural network is interesting for the cQA problem because: (*i*) it works in a pairwise fashion, i.e., given two translation hypotheses and a reference translation to compare to, the network decides which translation hypothesis is better, which is appropriate for a ranking problem; (*ii*) it allows for an easy incorporation of rich syntactic and semantic embedded representations of the input texts, and it efficiently models complex non-linear relationships between them; (*iii*) it uses a number of machine translation evaluation measures that have not been explored for the cQA task before, e.g., TER (Snover et al., 2006), METEOR (Lavie and Denkowski, 2009), and BLEU (Papineni et al., 2002).

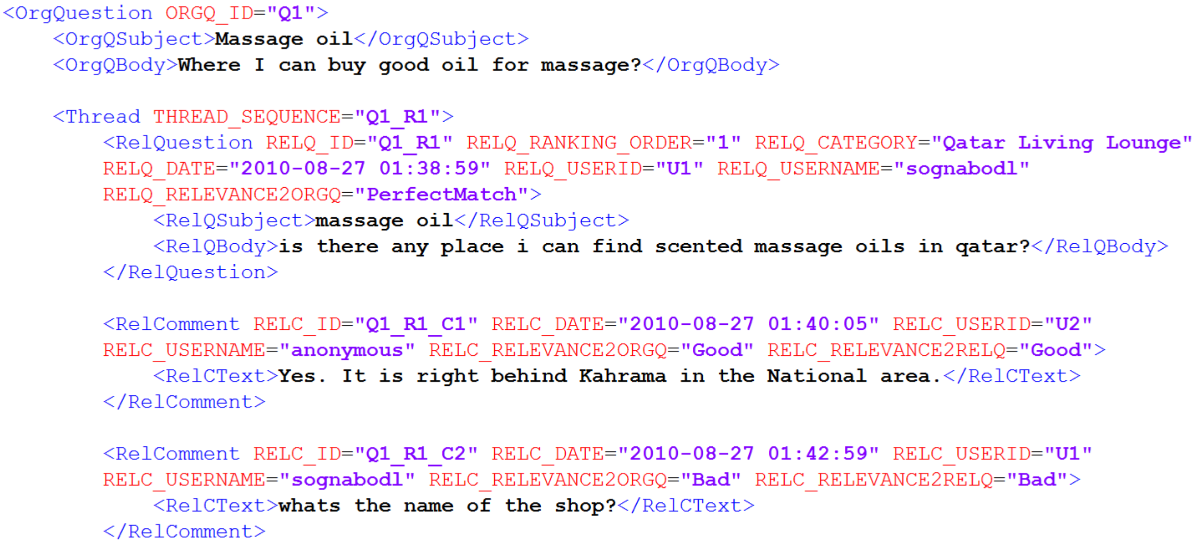
The analogy we apply to adapt the neural MTE architecture to the cQA problem is the following: given two comments c_1 and c_2 from the question thread—which play the role of the two competing translation hypotheses—we have to decide whether c_1 is a better answer than c_2 to question q—which plays the role of the translation reference. If we have a function $f(q, c_1, c_2)$ to make this decision, then we can rank the finite list of comments in the thread by comparing all possible pairs and by accumulating for each comment the scores for it given by f.

From a general point of view, MTE and the cQA task addressed in this paper seem similar: both reason about the similarity of two competing texts against a reference text in order to decide which one is better. However, there are some profound differences, which have implications on how each task is solved.

[1] SemEval-2016 Task 3 had two more subtasks: subtask B on Question-Question Similarity, and subtask C on Question-External Comment Similarity, which are out of our scope. However, they could be potentially addressed within our general MTE-NN framework, with minor variations.

[2] http://alt.qcri.org/semeval2016/task3/

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 460–466,
Berlin, Germany, August 7-12, 2016. ©2016 Association for Computational Linguistics



```
<OrgQuestion ORGQ_ID="Q1">
    <OrgQSubject>Massage oil</OrgQSubject>
    <OrgQBody>Where I can buy good oil for massage?</OrgQBody>

    <Thread THREAD_SEQUENCE="Q1_R1">
        <RelQuestion RELQ_ID="Q1_R1" RELQ_RANKING_ORDER="1" RELQ_CATEGORY="Qatar Living Lounge"
        RELQ_DATE="2010-08-27 01:38:59" RELQ_USERID="U1" RELQ_USERNAME="sognabodl"
        RELQ_RELEVANCE2ORGQ="PerfectMatch">
            <RelQSubject>massage oil</RelQSubject>
            <RelQBody>is there any place i can find scented massage oils in qatar?</RelQBody>
        </RelQuestion>

        <RelComment RELC_ID="Q1_R1_C1" RELC_DATE="2010-08-27 01:40:05" RELC_USERID="U2"
        RELC_USERNAME="anonymous" RELC_RELEVANCE2ORGQ="Good" RELC_RELEVANCE2RELQ="Good">
            <RelCText>Yes. It is right behind Kahrama in the National area.</RelCText>
        </RelComment>

        <RelComment RELC_ID="Q1_R1_C2" RELC_DATE="2010-08-27 01:42:59" RELC_USERID="U1"
        RELC_USERNAME="sognabodl" RELC_RELEVANCE2ORGQ="Bad" RELC_RELEVANCE2RELQ="Bad">
            <RelCText>whats the name of the shop?</RelCText>
        </RelComment>
```

Figure 1: Annotated English question from the CQA-QL corpus. Shown are the first two comments only.

In MTE, the goal is to decide whether a hypothesis translation conveys the same meaning as the reference translation. In cQA, it is to determine whether the comment is an appropriate answer to the question. Furthermore, in MTE we can expect shorter texts, which are typically much more similar. In contrast, in cQA, the question and the intended answers might differ significantly both in terms of length and in lexical content. Thus, it is not clear a priori whether the MTE network can work well to address the cQA problem. Here, we show that the analogy is not only convenient, but also that using it can yield state-of-the-art results for the cQA task.

To validate our intuition, we present series of experiments using the publicly available SemEval-2016 Task 3 datasets, with focus on subtask A. We show that a naïve application of the MTE architecture and features on the cQA task already yields results that are largely above the task baselines. Furthermore, by adapting the models with in-domain data, and adding lightweight task-specific features, we are able to boost our system to reach state-of-the-art performance.

More interestingly, we analyze the contribution of several features and parts of the NN architecture by performing an ablation study. We observe that every single piece contributes important information to achieve the final performance. While task-specific features are crucial, other aspects of the framework are relevant as well: syntactic embeddings, machine translation evaluation measures, and pairwise training of the network.

The rest of the paper is organized as follows: Section 2 introduces some related work. Section 3 presents the overall architecture of our MTE-inspired NN framework for cQA. Section 4 summarizes the features we use in our experiments. Section 5 describes the experimenal settings and presents the results. Finally, Section 6 offers further discussion and presents the main conclusions.

2 Related Work

Recently, many neural network (NN) models have been applied to cQA tasks: e.g., *question-question similarity* (Zhou et al., 2015; dos Santos et al., 2015; Lei et al., 2016) and *answer selection* (Severyn and Moschitti, 2015; Wang and Nyberg, 2015; Shen et al., 2015; Feng et al., 2015; Tan et al., 2015). Most of these papers concentrate on providing advanced neural architectures in order to better model the problem at hand. However, our goal here is different: we extend and reuse an existing pairwise NN framework from a different but related problem.

There is also work that uses machine translation models as a features for cQA (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) e.g., a variation of IBM model 1, to compute the probability that the question is a possible "translation" of the candidate answer. Unlike that work, here we port an entire MTE framework to the cQA problem. A preliminary version of this work was presented in (Guzmán et al., 2016).

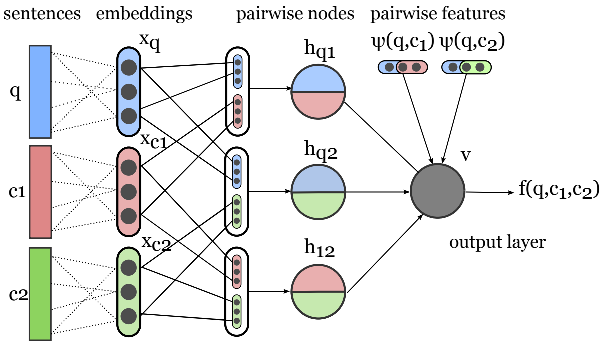


Figure 2: Overall architecture of the NN.

3 Neural Model for Answer Ranking

The NN model we use for answer ranking is depicted in Figure 2. It is a direct adaptation of our feed-forward NN for MTE (Guzmán et al., 2015). Technically, we have a binary classification task with input (q, c_1, c_2), which should output 1 if c_1 is a better answer to q than c_2, and 0 otherwise. The network computes a sigmoid function $f(q, c_1, c_2) = \mathrm{sig}(\mathbf{w_v^T}\phi(q, c_1, c_2) + b_v)$, where $\phi(x)$ transforms the input x 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.

We first map the question and the comments to a fixed-length vector $[\mathbf{x}_q, \mathbf{x}_{c_1}, \mathbf{x}_{c_2}]$ using syntactic and semantic embeddings. Then, we feed this vector as input to the neural network, which models three types of interactions, using different groups of nodes in the hidden layer. There are two *evaluation* groups $\mathbf{h_{q1}}$ and $\mathbf{h_{q2}}$ that model how good each comment c_i is to the question q. The input to these groups are the concatenations $[\mathbf{x}_q, \mathbf{x}_{c_1}]$ and $[\mathbf{x}_q, \mathbf{x}_{c_2}]$, respectively. The third group of hidden nodes $\mathbf{h_{12}}$, which we call *similarity* group, models how close c_1 and c_2 are. Its input is $[\mathbf{x}_{c_1}, \mathbf{x}_{c_2}]$. This might be useful as highly similar comments are likely to be comparable in appropriateness, irrespective of whether they are good or bad answers in absolute terms.

In summary, the transformation $\phi(q, c_1, c_2) = [\mathbf{h_{q1}}, \mathbf{h_{q2}}, \mathbf{h_{12}}]$ can be written as

$$\begin{aligned}\mathbf{h_{qi}} &= g(\mathbf{W_{qi}}[\mathbf{x_q}, \mathbf{x_{c_i}}] + \mathbf{b_{qi}}),\ i = 1, 2\\ \mathbf{h_{12}} &= g(\mathbf{W_{12}}[\mathbf{x_{c_1}}, \mathbf{x_{c_2}}] + \mathbf{b_{12}}),\end{aligned}$$

where $g(.)$ is a non-linear activation function (applied component-wise), $\mathbf{W} \in \mathbb{R}^{H \times N}$ are the associated weights between the input layer and the hidden layer, and $\mathbf{b}$ are the corresponding bias terms.

We use $\tanh$ as an activation function, rather than sig, to be consistent with how the word embedding vectors we use were generated.

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, skipping the hidden layer. These arcs represent pairwise *similarity* feature vectors between q and either c_1 or c_2. In these feature vectors, we encode MT evaluation measures (e.g., TER, METEOR, and BLEU), cQA task-specific features, etc. See Section 4 for detail about the features implemented as skip arcs. In the figure, we indicate these pairwise external feature sets as $\psi(q, c_1)$ and $\psi(q, c_2)$. When including the external features, the activation at the output is $f(q, c_1, c_2) = \mathrm{sig}(\mathbf{w_v^T}[\phi(q, c_1, c_2), \psi(q, c_1), \psi(q, c_2)] + b_v)$.

4 Features

We experiment with three kinds of features: (*i*) input embeddings, (*ii*) features from MTE (Guzmán et al., 2015) and (*iii*) task-specific features from SemEval-2015 Task 3 (Nicosia et al., 2015).

A. Embedding Features We used two types of vector-based embeddings to encode the input texts q, c_1 and c_2: (1) **GOOGLE_VECTORS**: 300-dimensional embedding vectors, trained on 100 billion words from Google News (Mikolov et al., 2013). The encoding of the full text is just the average of the word embeddings. (2) **SYNTAX**: 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.

Also, we compute cosine similarity features with the above vectors: $\cos(q, c_1)$ and $\cos(q, c_2)$.

B. MTE features We use the following MTE metrics (**MTFEATS**), which compare the similarity between the question and a candidate answer: (1) BLEU (Papineni et al., 2002); (2) NIST (Doddington, 2002); (3) TER v0.7.25 (Snover et al., 2006). (4) METEOR v1.4 (Lavie and Denkowski, 2009) with paraphrases; (5) Unigram PRECISION; (6) Unigram RECALL.

BLEUCOMP. We further use as features various components 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.

C. Task-specific features First, we train domain-specific vectors using WORD2VEC on all available QatarLiving data, both annotated and raw (**QL_VECTORS**).

Second, we compute various easy task-specific features (**TASK_FEATURES**), most of them proposed for the 2015 edition of the task (Nicosia et al., 2015). This includes some comment-specific features: (1) number of URLs/images/emails/phone numbers; (2) number of occurrences of the string "thank";[3] (3) number of tokens/sentences; (4) average number of tokens; (5) type/token ratio; (6) number of nouns/verbs/adjectives/adverbs/ pronouns; (7) number of positive/negative smileys; (8) number of single/double/triple exclamation/interrogation symbols; (9) number of interrogative sentences (based on parsing); (10) number of words that are not in WORD2VEC's Google News vocabulary.[4] Also some question-comment pair features: (1) question to comment count ratio in terms of sentences/tokens/nouns/verbs/adjectives/adverbs/pronouns; (2) question to comment count ratio of words that are not in WORD2VEC's Google News vocabulary. Finally, we also have two meta features: (1) is the person answering the question the one who asked it; (2) reciprocal rank of the comment in the thread.

5 Experiments and Results

We experiment with the data from SemEval-2016 Task 3. The task offers a higher quality training dataset TRAIN-PART1, which includes 1,412 questions and 14,110 answers, and a lower-quality TRAIN-PART2 with 382 questions and 3,790 answers. We train our model on TRAIN-PART1 with hidden layers of size 3 for 100 epochs with minibatches of size 30, regularization of 0.005, and a decay of 0.0001, using stochastic gradient descent with adagrad (Duchi et al., 2011); we use Theano (Bergstra et al., 2010) for learning. We normalize the input feature values to the $[-1; 1]$ interval using minmax, and we initialize the network weights by sampling from a uniform distribution as in (Bengio and Glorot, 2010). We train the model using all pairs of good vs. bad comments, in both orders, ignoring ties.

System	MAP	AvgRec	MRR
MTE-CQA$_{pairwise}$	**78.20**	**88.01**	**86.93**
MTE-CQA$_{classification}$	77.62	87.85	85.79
MTE$_{vanilla}$	70.17	81.84	78.60
Baseline$_{time}$	59.53	72.60	67.83
Baseline$_{rand}$	52.80	66.52	58.71

Table 1: Main results on the ranking task.

At test time, we get the full ranking by scoring all possible pairs, and we accumulate the scores at the comment level.

We evaluate the model on TRAIN-PART2 after each epoch, and ultimately we keep the model that achieves the highest accuracy;[5] in case of a tie, we prefer the parameters from an earlier epoch. We selected the above parameter values on the DEV dataset (244 questions and 2,440 answers) using the full model, and we used them for all experiments below, where we evaluate on the official TEST dataset (329 questions and 3,270 answers). We report mean average precision (MAP), which is the official evaluation measure, and also average recall (AvgRec) and mean reciprocal rank (MRR).

5.1 Results

Table 1 shows the evaluation results for three configurations of our MTE-based cQA system. We can see that the vanilla MTE system (MTE$_{vanilla}$), which only uses features from our original MTE model, i.e., it does not have any task-specific features (TASK_FEATURES and QL_VECTORS), performs surprisingly well despite the differences in the MTE and cQA tasks. It outperforms a random baseline (Baseline$_{rand}$) and a chronological baseline that assumes that early comments are better than later ones (Baseline$_{time}$) by large margins: by about 11 and 17 MAP points absolute, respectively. For the other two measures the results are similar.

We can further see that adding the task-specific features in MTE-CQA$_{pairwise}$ improves the results by another 8 MAP points absolute. Finally, the second line shows that adapting the network to do classification (MTE-CQA$_{classification}$), giving it a question and a single comment as input, yields a performance drop of 0.6 MAP points absolute compared to the proposed pairwise learning model. Thus, the pairwise training strategy is confirmed to be better for the ranking task, although not by a large margin.

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

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

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

System	MAP	AvgRec	MRR	Δ_{MAP}
MTE-CQA	**78.20**	**88.01**	**86.93**	
−BLEUCOMP	77.83	87.85	86.32	-0.37
−MTFEATS	77.75	87.76	86.01	-0.45
−SYNTAX	77.65	87.65	85.85	-0.55
−GOOGLE_VECT.	76.96	87.66	84.72	-1.24
−QL_VECTORS	75.83	86.57	83.90	-2.37
−TASK_FEATS.	72.91	84.06	78.73	-5.29

Table 2: Results of the ablation study.

Table 2 presents the results of an ablation study, where we analyze the contribution of various features and feature groups to the performance of the overall system. For the purpose, we study Δ_{MAP}, i.e., the absolute drop in MAP when the feature group is excluded from the full system.

Not surprisingly, the most important turn out to be the TASK_FEATURES (contributing over five MAP points) as they handle important information sources that are not available to the system from other feature groups, e.g., the reciprocal rank alone contributes about two points.

Next in terms of importance come word embeddings, QL_VECTORS (contributing over 2 MAP points), trained on text from the target forum, QatarLiving. Then come the GOOGLE_VECTORS (contributing over one MAP point), which are trained on 100 billion words, and thus are still useful even in the presence of the domain-specific QL_VECTORS, which are in turn trained on four orders of magnitude less data.

Interestingly, the MTE-motivated SYNTAX vectors contribute half a MAP point, which shows the importance of modeling syntax for this task. The other two MTE features, MTFEATS and BLEUCOMP, together contribute 0.8 MAP points. It is interesting that the BLEU components manage to contribute on top of the MTFEATS, which already contain several state-of-the-art MTE measures, including BLEU itself. This is probably because the other features we have do not model n-gram matches directly.

Finally, Table 3 puts the results in perspective. We can see that our system MTE-CQA would rank first on MRR, second on MAP, and fourth on AvgRec in SemEval-2016 Task 3 competition.[6] These results are also 5 and 16 points above the average and the worst systems, respectively.

System	MAP	AvgRec	MRR
1st (Filice et al., 2016)	79.19	88.82	86.42
MTE-CQA	***78.20***	***88.01***	***86.93***
2nd (Barrón-Cedeño et al., 2016)	77.66	88.05	84.93
3rd (Mihaylov and Nakov, 2016)	77.58	88.14	85.21
...	...	...	...
Average	73.54	84.61	81.54
...	...	...	...
12th (Worst)	62.24	75.41	70.58

Table 3: Comparative results with the best SemEval-2016 Task 3, subtask A systems.

This is remarkable given the lightweight task-specific features we use, and confirms the validity of the proposed neural approach to produce state-of-the-art systems for this particular cQA task.

6 Conclusion

We have explored the applicability of machine translation evaluation methods to answer ranking in community Question Answering, a seemingly very different task, where the goal is to rank the comments in a question-answer thread according to their appropriateness to the question, placing all good comments above all bad ones.

In particular, we have adopted a pairwise neural network architecture, which incorporates MTE features, as well as rich syntactic and semantic embeddings of the input texts that are non-linearly combined in the hidden layer. The evaluation results on benchmark datasets have shown state-of-the-art performance, with sizeable contribution from both the MTE features and from the network architecture. This is an interesting and encouraging result, as given the difference in the tasks, it was not a-priori clear that an MTE approach would work well for cQA.

In future work, we plan to incorporate other similarity measures and better task-specific features into the model. We further want to explore the application of this architecture to other semantic similarity problems such as question-question similarity, and textual entailment.

Acknowledgments

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

[6] The full results can be found on the task website: http://alt.qcri.org/semeval2016/task3/index.php?id=results

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, San Diego, CA.

Yoshua Bengio and Xavier Glorot. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, AISTATS '10, pages 249–256, 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 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, Texas, USA.

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, California, USA.

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.

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 Automatic Speech Recognition and Understanding Workshop*, ASRU '15, Scottsdale, Arizona, USA.

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, San Diego, California, USA.

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. 2016. 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, San Diego, California, USA.

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 Moschitti, and Lluís Màrquez. 2016. Semi-supervised question retrieval with gated convolutions. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '16, San Diego, California, USA.

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*, HLT '11, pages 1425–1434, Portland, Oregon, USA.

Todor Mihaylov and Preslav Nakov. 2016. 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, San Diego, California, USA.

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, Georgia, USA.

Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016. SemEval-2016 task 3: Community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, USA.

Massimo Nicosia, Simone Filice, Alberto Barrón-Cedeño, Iman Saleh, Hamdy Mubarak, Wei Gao, Preslav Nakov, Giovanni Da San Martino, Alessandro Moschitti, Kareem Darwish, Lluís Màrquez, Shafiq Joty, and Walid Magdy. 2015. QCRI: Answer selection for community question answering - experiments for Arabic and English. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '15, pages 203–209, Denver, Colorado, USA.

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 Meting of the Association for Computational Linguistics*, ACL '02, pages 311–318, Philadelphia, Pennsylvania, USA.

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.

Yikang Shen, Wenge Rong, Nan Jiang, Baolin Peng, Jie Tang, and Zhang Xiong. 2015. Word embedding based correlation model for question/answer matching. *arXiv preprint arXiv:1511.04646*.

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, Massachusetts, USA.

Richard Socher, John Bauer, Christopher D. Manning, and Ng Andrew Y. 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, March.

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, June.

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, Colorado, USA.

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.

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.