MTE-NN at SemEval-2016 Task 3: Can Machine Translation Evaluation Help 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 present a system for answer ranking (SemEval-2016 Task 3, subtask A) that is a direct adaptation of a pairwise neural network model for machine translation evaluation (MTE). In particular, the network incorporates MTE features, as well as rich syntactic and semantic embeddings, and it efficiently models complex non-linear interactions between them. With the addition of lightweight task-specific features, we obtained very encouraging experimental results, with sizeable contributions from both the MTE features and from the pairwise network architecture. We also achieved good results on subtask C.

1 Introduction

We present a system for SemEval-2016 Task 3 on Community Question Answering (cQA), subtask A (English). In that task, we are given a question from a community forum and a thread of associated text comments intended to answer the question, and the goal is to rank the comments according to their appropriateness to the question. Since cQA forum threads are noisy, as many comments are not answers to the question, the challenge lies in learning to rank all *good* comments above all *bad* ones.[1]

In this work, we approach subtask A from a novel perspective: by using notions of machine translation evaluation (MTE) to decide on the *quality* of a comment. In particular, we extend the MTE neural network framework from Guzmán et al. (2015). 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; this 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 among them; (*iii*) it uses a number of MT evaluation measures that have not been explored for the cQA task (e.g., TER, Meteor and BLEU).

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 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.

The two tasks seem similar: both reason about the similarity of two competing texts against a reference text, to decide which one is better. However, there are some profound differences. 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 much more similar among them. In cQA, the question and the intended answers might differ significantly both in length and in lexical content. Thus, it is not clear a priori whether the MTE network can work well for cQA. Here, we show that the analogy is convenient, allowing to achieve competitive results.

[1] More detail and examples can be found on the task website (http://alt.qcri.org/semeval2016/task3/) and in the associated task description paper (Nakov et al., 2016).

Proceedings of SemEval-2016, pages 812–820,
San Diego, California, June 16-17, 2016.

At competition time, we achieved the sixth best result on the task from a set of twelve systems. Right after the competition we introduced some minor improvements and extra features, without changing the fundamental architecture of the network, which improved the MAP result by almost two points. We also performed a more detailed experimental analysis of the system, checking the contribution of several features and parts of the NN architecture. We observed 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 too: syntactic embeddings, MT evaluation measures, and pairwise training of the network.

Finally, we used our system for subtask A to solve subtask C, which asks to find good answers to a new question that was not asked before in the forum by reranking the answers to related questions. For the purpose, we weighted the subtask A scores by the reciprocal rank of the related questions (following the order given by the organizers, i.e., the ranking by Google). Without any subtask C specific addition, we achieved the fourth best result in the task.

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). Also, other participants in the SemEval 2016 Task 3 applied NNs to solve some of the subtasks (Nakov et al., 2016). However, our goal was different: we were interested in extending an existing pairwise NN framework from a different but related problem.

There is also work that uses scores from 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 "translation" of the candidate answer. Unlike that work, here we use machine translation *evaluation* (MTE) instead of machine translation models.

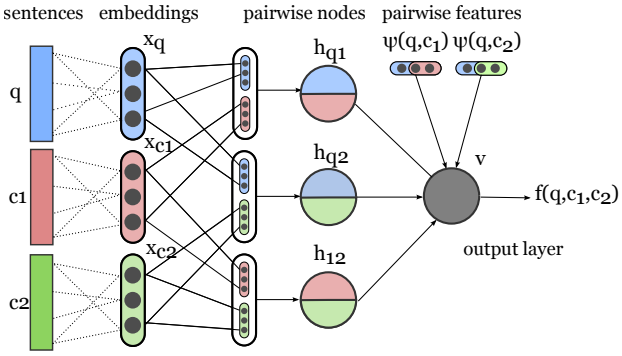

Figure 1: Overall architecture of the NN.

Another relevant work is that of Madnani et al. (2012), who applied MTE metrics as features for paraphrase identification. However, here we have a different problem: cQA. Moreover, instead of using MTE metrics as features, we port an entire MTE framework to the cQA problem.

3 Neural Model for Answer Ranking

The NN model we use for answer ranking is depicted in Figure 1. It is a direct adaptation of the feed-forward NN for MTE described in (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.[2] The network computes a sigmoid function $f(q, c_1, c_2) = \text{sig}(\mathbf{w_v^T}\phi(q, c_1, c_2) + b_v)$, 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 $\mathbf{h_{q_1}}$ and $\mathbf{h_{q_2}}$ 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.

[2] In this work, we do not learn to predict ties.

In summary, the transformation $\phi(q, c_1, c_2) = [\mathbf{h_{q_1}}, \mathbf{h_{q_2}}, \mathbf{h_{12}}]$ can be written as follows:

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

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 parts of our input vectors (the word embeddings) are 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 4.3 for details 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 Learning Features

We experiment with three kinds of features: (*i*) input embeddings, (*ii*) features motivated by previous work on Machine Translation Evaluation (MTE) (Guzmán et al., 2015) and (*iii*) task-specific features, mostly proposed by participants in the 2015 edition of the task (Nakov et al., 2015).

4.1 Embedding Features

We use the following vector-based embeddings of (q, c_1, c_2) as input to the NN:

- GOOGLE_VEC: We use the pre-trained, 300-dimensional embedding vectors, which Tomas Mikolov trained on 100 billion words from Google News (Mikolov et al., 2013).
- SYNTAX_VEC: We parse the entire question/comment text 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. More specifically, given a question q and a pair of comments c_1 and c_2 for it, we calculate the following features: $\psi(q, c_1) = cos(q, c_1)$ and $\psi(q, c_2) = cos(q, c_2)$.

4.2 MTE features

MTFEATS (in MTE-NN-improved only). We use (as skip-arc pairwise features) the following six machine translation evaluation features, to which we refer as MTFEATS, and which measure the similarity between the question and a candidate answer:

- 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).
- NIST: This measure is similar to BLEU, and is used at evaluation campaigns run by NIST (Doddington, 2002).
- TER: Translation error rate; it is based on the edit distance between a translation hypothesis and the reference (Snover et al., 2006).
- METEOR: A measure that matches the hypothesis and the reference using synonyms and paraphrases (Lavie and Denkowski, 2009).
- PRECISION: measure, originating in information retrieval.
- RECALL: another measure coming from information retrieval.

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. We will refer to the set of these features as BLEUCOMP.

4.3 Task-specific features

QL_VEC (in MTE-NN-improved only). Similarly to the GOOGLE_VEC, but on task-specific data, we train word vectors using WORD2VEC on all available cQA training data (Qatar Living) and use them as input to the NN.

QL+IWSLT_VEC (in MTE-NN-{primary, contrastive1/2} only). We also use trained word vectors on the concatenation of the cQA training data and the English portion of the IWSLT data, which consists of TED talks (Cettolo et al., 2012) and is thus informal and somewhat similar to cQA data.

TASK_FEAT. We further extract various task-specific skip-arc features, most of them proposed for the 2015 edition of the task (Nakov et al., 2015). This includes some comment-specific features:

- number of URLs/images/emails/phone numbers;
- number of occurrences of the string *thank*;[3]
- number of tokens/sentences;
- average number of tokens;
- type/token ratio;
- 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.[4]

And also some question-comment pair features:

- question to comment count ratio in terms of sentences/tokens/nouns/verbs/adjectives/adverbs/pronouns;
- question to comment count ratio of words that are not in word2vec's Google News vocabulary.

We also have two meta features:

- is the person answering the question the one who asked it;
- reciprocal rank of the comment in the thread.

[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 Experiments and Results

Below we explain which part of the available data we used for training, as well as our basic settings. Then, we present in detail our experiments and the evaluation results.

5.1 Data and Settings

We experiment with the data from SemEval-2016 Task 3 (Nakov et al., 2016). 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 63 epochs with minibatches of size 30, regularization of 0.0015, 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 and bad comments, ignoring ties. At test time we get the full ranking by scoring all possible pairs, and accumulating 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 Kendall's Tau (τ); in case of a tie, we prefer the parameters from a later epoch. We selected the above parameter values on the DEV dataset (244 questions and 2,440 answers) using the full model, and we use them for all experiments below, where we evaluate on the official TEST dataset (329 questions and 3,270 answers).

For evaluation, we use mean average precision (MAP), which is the official evaluation measure. We further report scores using average recall (AvgRec), mean reciprocal rank (MRR), Precision (P), Recall (R), F-measure (F_1), and Accuracy (Acc). Note that the first three are ranking measures, to which we directly give our ranking scores. However, the latter four measures require Good vs. Bad categorical predictions. We generate them based on the ranking scores using a threshold: if the score is above 0.95 (chosen on the DEV set), we consider the comment to be Good, otherwise it is Bad.

5.2 Contrastive Runs

We submitted two contrastive runs, which differ from the general settings above as follows:

- MTE-NN-contrastive1: a different network architecture with 50 units in the hidden layer (instead of 3 for each of $\mathbf{h_{q1}}$,$\mathbf{h_{q2}}$,$\mathbf{h_{12}}$) and higher regularization (0.03, i.e., twenty times bigger). On the development data, it performed very similarly to those for the primary run, and we wanted to try a bigger NN.

- MTE-NN-contrastive2: the same architecture as the primary but different training. We put together TRAIN-PART1 and DEV and randomly split them into 90% for training and 10% for model selection. The idea here was to have some training examples from development, which was supposed to be a cleaner dataset (and so more similar to the test set).

5.3 Official Results

Table 1 shows the results for our submissions for subtask A. Our primary submission was ranked sixth out of twelve teams on MAP. Note, however, that it was third on MRR and F_1. It is also 3 and 14 points above the average and the worst systems, respectively, and well above the baselines. Both our contrastive submissions performed slightly better, but neither of them is strong enough to change the overall ranking if we had chosen one of them as primary.

For subtask C, we multiplied (*i*) our scores for subtask A for the related question by (*ii*) the given reciprocal rank of the related question in the list of related questions. That is, we did not try to address question-question similarity (subtask B). We achieved 4th place with a MAP of 49.38, which is well above the baseline of 40.36. Our contrastive2 run performed slightly better at 49.49.

5.4 Post-submission Analysis on the Test Set

After the competition, we produced a refined version of the system (**MTE-NN-improved**) where the settings changed as follows: (*i*) using QL_VEC instead of QL+IWSLT_VEC, (*ii*) adding MTFEATS to the set of features, (*iii*) optimizing accuracy instead of Kendall's tau, (*iv*) training for 100 epochs instead of 63, and (*v*) regularization of 0.005 instead of 0.0015.

System	**MAP**	AvgRec	MRR	Δ_{MAP}
MTE-NN-improved	**78.20**	**88.01**	**86.93**	
−TASK_FEATS	72.91	84.06	78.73	-5.29
−COMMENT_RANK	76.08	86.41	84.42	-2.12
−SAME_AUTHOR	76.60	86.75	83.71	-1.60
−QL_VEC	75.83	86.57	83.90	-2.37
−GOOGLE_VEC	76.96	87.66	84.72	-1.24
−SYNTAX_VEC	77.65	87.65	85.85	-0.55
−COSINES	76.97	87.28	85.03	-1.23
−MTFEATS	77.75	87.76	86.01	-0.45
−BLEUCOMP	77.83	87.85	86.32	-0.37

Table 2: Ablation study of our improved system on the test data.

Note that the training and development set remained unchanged. **MTE-NN-improved** showed notable improvements on the DEV set over our primary submission. In Table 2, we present the results on the TEST set. To gain additional insight about the contribution of various features and feature groups to the performance of the overall system, we also present the results of an ablation study where we removed different feature groups one by one. For this purpose, we study Δ_{MAP}, i.e., the absolute change in MAP when the feature or feature group is excluded from the full system. Not surprisingly, the most important turn out to be the TASK_FEATS (contributing over 5 MAP points) as they handle important information sources that are not available to the system from other feature groups, e.g., the reciprocal rank of the comment in the comment thread, which alone contributes 2.12 MAP points, and the feature checking whether the person who asked the question is the one who answered, which contributes 1.60 MAP points. Next in terms of importance come word embeddings, QL_VEC (contributing over 2 MAP points), trained on text from the target forum, QatarLiving. Then come the GOOGLE_VEC (contributing over 1 MAP point), which are trained on 100 billion words, and thus are still useful even in the presence of the domain-specific QL_VEC, which are in turn trained on four orders of magnitude less data. Interestingly, the MTE-motivated SYNTAX_VEC vectors contribute half a MAP point, which shows the importance of modeling syntax for this task. Next, we can see that using just the vectors is not enough, and adding cosines as pairwise features for the three kinds of vectors contributes over one MAP point.

	Submission	MAP	AvgRec	MRR	P	R	F1	Acc
1	SemEval 1st	79.19_1	88.82_1	86.42_1	76.96_1	55.30_8	64.36_5	75.11_2
	MTE-NN-improved	***78.20***	***88.01***	***86.93***	***57.08***	***76.75***	***65.47***	***67.09***
2	SemEval 2nd	77.66_2	88.05_3	84.93_4	75.56_2	58.84_6	66.16_2	75.54_1
3	SemEval 3rd	77.58_3	88.14_2	85.21_2	74.13_4	53.05_{10}	61.84_8	73.39_5
4	SemEval 4th	77.28_4	87.52_5	84.09_6	70.46_6	63.36_4	66.72_1	74.31_4
5	SemEval 5th	77.16_5	87.98_4	84.69_5	74.43_3	56.73_7	64.39_4	74.50_3
	MTE-NN-contrastive2	*76.98*	*86.98*	*85.50*	*58.71*	*70.28*	*63.97*	*67.83*
	MTE-NN-contrastive1	*76.86*	*87.03*	*84.36*	*55.84*	*77.35*	*64.86*	*65.93*
6	**MTE-NN-primary**	$\mathbf{76.44_6}$	$\mathbf{86.74_7}$	$\mathbf{84.97_3}$	$\mathbf{56.28_9}$	$\mathbf{76.22_1}$	$\mathbf{64.75_3}$	$\mathbf{66.27_8}$
...	...	...	...	...	...	...	...	...
	SemEval Average	73.54	84.61	81.54	64.80	57.03	58.77	68.45
...	...	...	...	...	...	...	...	...
12	SemEval 12th	62.24_{12}	75.41_{12}	70.58_{12}	50.28_{11}	53.50_9	51.84_{10}	59.60_{11}
	Baseline 1 (IR)	**59.53**	**72.60**	**67.83**	—	—	—	—
	Baseline 2 (random)	52.80	66.52	58.71	40.56	74.57	52.55	45.26
	Baseline 3 (all 'true')	—	—	—	40.64	100.00	**57.80**	40.64
	Baseline 4 (all 'false')	—	—	—	—	—	**—**	**59.36**

Table 1: **Comparison to the official results on SemEval-2016 Task 3, subtask A.** The first column shows the rank of the primary runs with respect to the official MAP score. The subindices in the results columns show the rank of the primary runs with respect to the evaluation measure in the respective column.

Finally, the 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.

We further used the output of our MTE-NN-improved system to generate predictions for subtask C, as explained above. This yielded improvements from 49.38 to 49.87 on MAP, from 55.44 to 56.08 on AvgRec, and from 51.56 to 52.16 on MRR.

6 Conclusion

We have explored the applicability of machine translation evaluation metrics to answer ranking in community Question Answering, a seemingly very different task (compared to MTE). In particular, with ranking in mind, 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.

Our post-competition improvements have shown state-of-the-art performance (Guzmán et al., 2016), with sizeable contribution from both the MTE features and from the network architecture. This is an encouraging result as it was not a priori clear that an MTE approach would work well for cQA.

In future work, we plan to incorporate fine-tuned word embeddings as in the SemanticZ system (Mihaylov and Nakov, 2016b), and information from entire threads (Nicosia et al., 2015; Barrón-Cedeño et al., 2015; Joty et al., 2015; Joty et al., 2016). We also want to add more knowledge sources, e.g., as in the SUper Team system (Mihaylova et al., 2016), including veracity, sentiment, complexity, troll user features as inspired by (Mihaylov et al., 2015a; Mihaylov et al., 2015b; Mihaylov and Nakov, 2016a), and PMI-based goodness polarity lexicons as in the PMI-cool system (Balchev et al., 2016).

We further plan to explore the application of our NN architecture to subtasks B and C, and to study the interactions among the three subtasks in order to solve the primary subtask C. Furthermore, we would like to try a similar neural network for 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.

References

Daniel Balchev, Yasen Kiprov, Ivan Koychev, and Preslav Nakov. 2016. PMI-cool at SemEval-2016 Task 3: Experiments with pmi and goodness polarity lexicons for community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, USA.

Alberto Barrón-Cedeño, Simone Filice, Giovanni Da San Martino, Shafiq Joty, Lluís Màrquez, Preslav Nakov, and Alessandro Moschitti. 2015. Thread-level information for comment classification in community question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, ACL-IJCNLP '15, pages 687–693, Beijing, China.

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*, volume 9 of *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 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.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation*, EAMT '12, pages 261–268, Trento, Italy.

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 (Volume 2: Short Papers)*, 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 - Volume 1*, 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.

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 (Volume 1: Long Papers)*, ACL-IJCNLP '15, pages 805–814, Beijing, China.

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

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.

Shafiq Joty, Alberto Barrón-Cedeño, Giovanni Da San Martino, Simone Filice, Lluís Màrquez, Alessandro Moschitti, and Preslav Nakov. 2015. Global thread-level inference for comment classification in community question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, EMNLP '15, pages 573–578, Lisbon, Portugal.

Shafiq Joty, Lluís Màrquez, and Preslav Nakov. 2016. Joint learning with global inference for comment classification in community question answering. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '16, San Diego, California, USA.

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. Denoising bodies to titles: Retrieving similar questions with recurrent convolutional models. 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 - Volume 1*, HLT '11, pages 1425–1434, Portland, Oregon, USA.

Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '12, pages 182–190, Montréal, Canada.

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, 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, San Diego, California, USA.

Todor Mihaylov, Georgi Georgiev, and Preslav Nakov. 2015a. 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.

Todor Mihaylov, Ivan Koychev, Georgi Georgiev, and Preslav Nakov. 2015b. Exposing paid opinion manipulation trolls. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, RANLP '15, pages 443–450, Hissar, Bulgaria.

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, 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, Walid Magdy, Alessandro Moschitti, Jim Glass, and Bilal Randeree. 2015. Semeval-2015 task 3: Answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '15, pages 269–281, Denver, Colorado, USA.

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 (Volume 1: Long Papers)*, 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 (Volume 2: Short Papers)*, 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 (Volume 2: Short Papers)*, ACL-IJCNLP '15, pages 250–259, Beijing, China.