Learning Semantic Relatedness in Community Question Answering Using Neural Models

Henry Nassif, Mitra Mohtarami, James Glass
MIT Computer Science and Artificial Intelligence Laboratory
Cambridge, MA 02139, USA
{hnassif, mitram, glass}@mit.edu

Abstract

Community Question Answering forums, such as Quora and Stackoverflow contain millions of questions and answers. Automatically finding the relevant questions from the existing questions and finding the relevant answers to a new question are Natural Language Processing tasks. In this paper, we aim to address these tasks, which we refer to as *similar-Question Retrieval* and *Answer Selection*. We present a neural-based model with stacked bidirectional LSTMs and MLP to address these tasks. The model generates the vector representations of the question-question or question-answer pairs and computes their semantic similarity scores, which are then employed to rank and predict relevancies. Extensive experiments demonstrate our results outperform the baselines.

1 Introduction

Community Question Answering (cQA) websites such as Quora[1] and Stackoverflow[2] are rapidly expanding. Managing such platforms has become increasingly difficult because of the exponential growth in content, triggered by wider access to the internet. Traditionally, websites used to keep track of a list of frequently asked questions (FAQ) that they expect visitors to consult before asking a question. Now, with a wider range of questions being asked, a need has emerged for a better and more scalable system to automatically identify similarities between any two questions on the platform. In addition, with many users contributing to a single question, it has become harder to identify which answers are more relevant than others. We summarize these two problems as follows:

- *Question Retrieval:* given a new question and a list of questions, we automatically rank the questions in the list according to their relevancy to the new question.
- *Answer Selection:* given a cQA thread containing a question and a list of answers, we automatically rank the answers according to their relevance to the question.

The increase in the number of community-based Q&A platforms has lead to a rapid build up of large archives of user-generated questions and answers. When a new question is asked on the platform, the system searches for questions that are semantically similar in the archives. If a similar question is found, the corresponding correct answer is retrieved and returned immediately to the user as the final answer. The quality of the answer depends on the effectiveness of the question-similarity calculation. However, measuring semantic relatedness between questions and answers is not trivial. Sometimes, similar questions or relevant answers use very different wording. For instance, the two questions "Is downloading movies illegal?" and "Can I share a copy of a DVD online" have an almost identical meaning but are lexically very different. Traditional text-based similarity metrics for measuring sentence distance such as the Jaccard coefficient and the overlap coefficient (Manning and Schütze, 1999), perform poorly. In this paper, we present a neural-based model including stacked Bidirectional Long Short-Term Memory (BLSTM) networks and Multi-Layer Perceptron (MLP) to address the question retrieval and answer selection problems. The model computes the representations of the Q&As and then their semantic simi-

[1] https://www.quora.com/
[2] http://stackexchange.com/

Proceedings of the 1st Workshop on Representation Learning for NLP, pages 137–147,
Berlin, Germany, August 11th, 2016.

larity scores. These scores are subsequently employed to rank the list of existing questions and answers with respect to the given question. We evaluate our model on a public benchmark cQA data (Nakov et al., 2016), and show that the results of our model outperform the baselines.

2 Related Work

2.1 Question Retrieval

As explained in Section 1, two questions that are worded very differently can be similar in meaning. Three types of approaches have been developed in the literature to solve this word mismatch problem among similar questions. The first type of approach uses knowledge databases such as dictionaries. For example, Frequently Asked Question (FAQ) Finder (Burke et al., 1997) heuristically combined statistical similarities computed using conventional vector space models with semantic similarities between questions estimated using WordNet (Fellbaum, 1998) to rank FAQs. Song et al. (2007) presented an approach which is a linear combination of statistic similarity, calculated based on word co-occurrence, and semantic similarity, calculated using WordNet and a bipartite mapping. Auto-FAQ (Whitehead, 1995) applied shallow language understanding into automatic FAQ answering, where the matching of a question to FAQs is based on keyword comparison enhanced by limited language processing techniques. However, the quality and structure of current knowledge databases are, based on the results of previous experiments, not good enough for reliable performance.

The second type of approach employed manual rules or templates. These methods are expensive and hard to scale for large size collections. Sneiders (2002) proposed template based FAQ retrieval systems, while Kim and Seo (2006) proposed using user click logs to find similar queries. Lai et al. (2002) proposed an approach to automatically mine FAQs from the web; However, they did not study the use of these FAQs after they were collected. Berger et al. (2000) proposed a statistical lexicon correlation method. These previous approaches were tested with relatively small sized collections and are hard to scale because they are based on specific knowledge databases or handcrafted rules.

The third type of approach uses statistical techniques developed in information retrieval and natural language processing (Berger et al., 2000). Jeon et al. (2005) presented question retrieval methods that are based on using the similarity between answers in the archive to estimate probabilities for a translation-based retrieval model. They run the IBM model 1 (Brown et al., 1993) to learn word translation probabilities on a collection of question pairs. Given a new question, a translation based information retrieval model exploits the word relationships to retrieve similar questions from Q&A archives. They show that with this model it is possible to find semantically similar questions with relatively little word overlap.

2.2 Answer Selection

Passage reordering or reranking has always been an essential step of automatic answer selection (Radlinski and Joachims, 2005; Jeon et al., 2005; Shen and Lapata, 2007; Moschitti et al., 2007; Severyn and Moschitti, 2015a; Moschitti, 2008; Tymoshenko and Moschitti, 2015; Surdeanu et al., 2008). Many methods have been proposed, such as exploring web redundancy information for answer validation (Magnini et al., 2002) and using non-textual features (Jeon et al., 2006).

Recently, many advanced models have been developed for automating answer selection based on syntactic structures (Severyn and Moschitti, 2012; Severyn and Moschitti, 2013; Grundström and Nugues, 2014) and textual entailment. These models include quasi-synchronous grammar to learn syntactic transformations from the question to the candidate answers (Wang et al., 2007); Continuous word and phrase vectors to encode semantic similarity (Belinkov et al., 2015); Tree Edit Distance (TED) to learn tree transformations in pairs (Heilman and Smith, 2010); probabilistic model to learn tree-edit operations on dependency parse trees (Wang and Manning, 2010); and linear chain CRFs with features derived from TED to automatically learn associations between questions and candidate answers (Yao et al., 2013).

In addition to the usual local features that only look at the question-answer pair, automatic answer selection algorithms can rely on global thread-level features, such as the position of the answer in the thread (Hou et al., 2015), or the context of an answer in a thread (Nicosia et al., 2015), or dependencies between thread answers using structured prediction models (Barrón-Cedeno et al., 2015).

Joty et al. (2015) modeled the relations between

pairs of answers at any distance in the thread, which they combine in a graph-cut and in an Integer Linear Programming (ILP) frameworks. They then proposed a fully connected pairwise CRFs (FCCRF) with global normalization and an Ising-like edge potential.

2.3 Neural Networks

Neural based approaches have wide applications including speech recognition (Graves and Jaitly, 2014), language modeling (Mikolov et al., 2010; Mikolov et al., 2011; Sutskever et al., 2011), translation (Liu et al., 2014; Sutskever et al., 2014; Auli et al., 2013), and image captioning (Karpathy and Fei-Fei, 2015). In addition, recent work shows the effectiveness of neural models in answer selection (Severyn and Moschitti, 2015b; Tan et al., 2015; Feng et al., 2015) and question similarity (dos Santos et al., 2015) in community question answering.

Dos Santos et al. (2015) developed CNN and bag-of-words (BOW) representation models for the question similarity task. Cosine similarity between the representations of the input questions were used to compute the CNN and BOW similarity scores for the question-question pairs. The convolutional representations in conjunction with other vectors are then passed to a MLP to compute the similarity score of the question pair. Furthermore, recent research has shown the effectiveness of CNNs for answer ranking of *short* textual contents (Severyn and Moschitti, 2015b).

In this paper, we present a neural model based on stacked bidirectional LSTMs and MLP to capture the long dependencies in longer-length questions and answers.

3 Method

In this paper, we present a neural based model using stacked bidirectional LSTMs and MLP to address the question retrieval and answer selection problems. We first briefly explain recurrent neural networks (RNNs), Long Short-Term Memory (LSTM) networks and their bidirectional networks. Then, we present the stacked bidirectional LSTMs for capturing the semantic similarity of questions and answers in cQA.

Recurrent Neural Networks: A recurrent neural network (RNN) has the form of a chain of repeating modules of neural network. This architecture is pertinent to learning sequences of information because it allows information to persist across states. The output of each loop is utilized as input to the following loop through hidden states that capture information about the preceding sequence.

RNNs are trained using backpropagation through time (BPTT) where the gradient at each output depends on the current and previous time steps. The BPTT approach is not effective at learning long term dependencies because of the exploding gradients problem (Pascanu et al., 2012; Bengio et al., 1994). A certain type of RNN, Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) has been designed to improve the learning of long-term dependencies.

Long Short-Term Memory Recurrent Neural Networks: Similar to Recurrent Neural Networks, Long Short-Term Memory Networks (LSTM) (Hochreiter and Schmidhuber, 1997) have a chain like architecture, with a different module structure. Instead of having a single neural network layer, each module has four layers filling different purposes. Each LSTM unit contains a memory cell with self-connections, as well as three multiplicative gates - forget, input, output - to control information flow. Each gate is composed of a sigmoid neural net layer and a pointwise multiplication operation.

Given input vector x_t, previous hidden outputs h_{t-1}, and previous cell state c_{t-1}, the LSTM unit performs the following operations:

$$
\begin{aligned}
f_t &= \sigma(W_f.[h_{t-1}, x_t] + b_f) \\
i_t &= \sigma(W_i.[h_{t-1}, x_t] + b_i) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(W_c.[h_{t-1}, x_t] + b_c) \\
o_t &= \sigma(W_o.[h_{t-1}, x_t] + b_o) \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

where f_t, i_t, o_t and h_t respectively represent the forget gate, input gate, output gate and the hidden layer.

Many variants of LSTMs were later introduced, such as depth gated RNNs (Yao et al., 2015), clockwork RNNS (Koutnik et al., 2014), and Gated Recurrent Unit RNNs (Cho et al., 2014).

Bidirectional Recurrent Neural Networks: Bidirectional RNNs (Schuster and Paliwal, 1997) or BRNN use past and future context sequences to predict or label each element. This is done by combining the outputs of two RNN, one

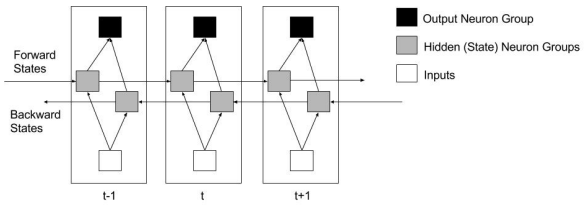

Figure 1: Bidirectional Long Short-Term Recurrent Neural Network. Bidirectional LSTMs are equivalent to two LSTMs independently updating their parameters by processing the input either in forward or backward direction.

processing the sequence forward (or left to right), the other one processing the sequence backwards (from right to left) as shown in Figure 1. This technique proved to be especially useful when combined with LSTM (Graves and Schmidhuber, 2005).

3.1 Stacked Bidirectional LSTMs for cQA

Given a question, we aim to rank a list of questions for question retrieval and a list of answers for answer selection. To address these ranking problems, we propose a neural model to compute the semantic similarities for the question-question (q, q') or question-answer (q, a) pairs. These scores are then employed to rank the list of questions and answers with respect to the given question q. Figure 2 shows the general architecture of our model. We explain our model by referring to the pair (q, a), but the same applies to the pair (q, q'). The question q and answer a contain the lists of words:

$$q = \{w_1^q, w_2^q, w_3^q, ..., w_k^q\}$$
$$a = \{w_1^a, w_2^a, w_3^a, ..., w_m^a\}$$

where w_i^q and w_i^a are the i^{th} word of the q and a respectively.

First, the q and a are truncated to have similar length[3], and two lists of vectors corresponding to the words for the question q and a are generated and randomly initialized:

$$V_q = \{X_1, X_2, X_3, ..., X_{n/2}\}$$
$$V_a = \{X_{n/2+1}, X_{n/2+2}, X_{n/2+3}, ..., X_n\}$$

where X_i with $i \in [1, n/2]$ is the vector of w_i^q for the q, X_i with $i \in [n/2+1, n]$ is the vector of $w_{i-n/2}^a$ for the a[4].

[3]We truncate the length of questions and answers to a maximum of 100 words. The questions and answers with less than 100 words are padded with zeros.

[4]n equals to 200

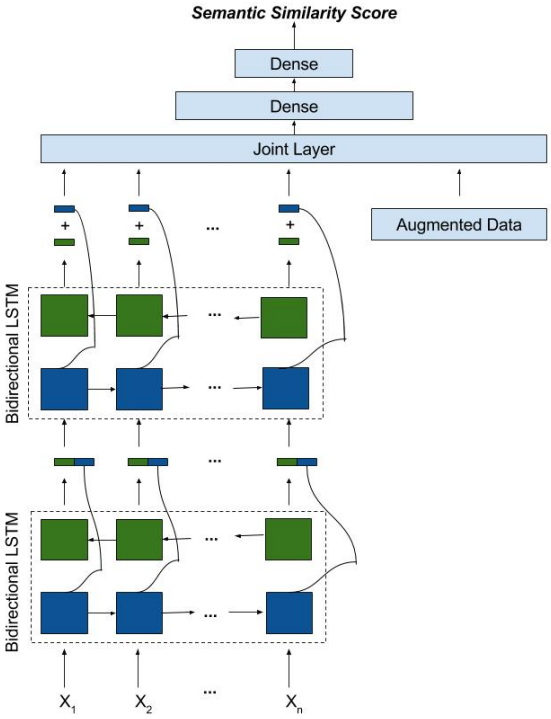

Figure 2: The general architecture of our model including the stacked Bidirectional LSTMs and MLP. The model is built on two bidirectional LSTMs whose output can be augmented with extra features and fed into the multi-layer perceptron.

The word vectors for the q (i.e., V_q) are passed to the model as shown in Figure 2. The model computes the representation of the question q after passing its last word vector to the model. Then the q representation along with the word vectors of the answer a (i.e., V_a) are passed to the model. The model generates the representation of the given pair (q, a) after processing the last word vector of the answer a affected by the representation of q. This information processing is performed at the forward layer of the first bidirectional LSTM shown in the figure (left to right). Similar processing in the reverse direction (right to left) is further applied on the given pair at the first bidirectional LSTM. The output vectors of the hidden layers for these two directions of the first bidirectional LSTM are then concatenated and inputted into the second bidirectional LSTM as shown in the Figure 2.

While the second bidirectional LSTM processes the input vectors similarly to the first one, its output vectors from two directions are summed[5] instead of concatenated. Finally, the resulting vec-

[5]Using summation instead of concatenation is selected based on the experimental results on the development set.

Embedding	initialized, updated
Weights for Two LSTMs	not shared
Optimizer	Adam
Learning rate	0.001
Dropout rate	0.5
Batch Size	16

Table 1: The hyper-parameters of the stacked bidirectional LSTM model.

Category	Train	Dev	Test
New Coming Questions	267	50	70
Related Questions	2,669	500	700
– Perfect-Match	235	59	81
– Relevant	848	155	152
– Irrelevant	1,586	286	467
Related Answers	17,900	2,440	7,000
– Good	6651	818	2,767
– Bad	8,139	1,209	3,090
– Potentially-Useful	3,110	413	1,143

Table 2: The statistics for the cQA data (Nakov et al., 2016) that we employ to evaluate our neural model.

tors can be augmented with the additional features and passed to the MLP with two hidden layers in order to compute the semantic similarity score of the q and a.

4 Results and Discussion

Hyper-parameters: Table 1 shows the hyper-parameters used in our model. The values for the hyper-parameters are optimized with respect to the results on the development set. The word vectors are randomly initialized and updated during the training step as explained in Section 3, and the weights for the two bidirectional LSTMs of the model are not shared. We employ *Adam* (Kingma and Ba, 2014) as the optimization method and *mean squared error* as loss function for our model. We further use the values 0.001, 0.5 and 16 for learning rate, dropout rate and batch size respectively.

Dataset: We evaluate our model on the cQA data (Nakov et al., 2016) in which the questions and answers have been manually labeled by a community of annotators in a crowdsourcing platform. Table 2 shows the statistics for the train, development and test data. The related questions are labeled as *Perfect-Match*, *Relevant* and *Irrelevant* with respect to an original question in the question retrieval task. The *Irrelevant* questions should be ranked lower than the other questions by the model. In addition, the answers are labeled as *Good*, *Bad* and *Potentially-Useful* with respect to a question in the answer selection task. The

Text-based features
– Longest Common Substring
– Longest Common Subsequence
– Greedy String Tiling
– Monge Elkan Second String
– Jaro Second String
– Jaccard coefficient
– Containment similarity
Vector-based features
– Normalized Averaged Word Vectors using `word2vec` (Mikolov et al., 2013)
– Most similar sentence pair for a given (q, a) using sentence vector representation
– Most similar chunk pair for a given (q, a) using chunk vector representation
Metadata-based features
– User information, like user id

Table 3: Some of the most important text- and vector- based features employed in the Bag-of-Vectors (BOV) baseline (Belinkov et al., 2015).

expected result is that both *Good* and *Potentially-Useful* answers have useful information, while the *Good* answers should be ranked higher than both *Potentially-Useful* and *Bad* answers.

Baselines: We compare our neural model with the BOV, BM25, IR and TF-IDF baselines that are briefly explained below:

- *Bag-of-Vectors (BOV):* This baseline employed various text- and vector- based features for the cQA problems (Belinkov et al., 2015). We highlight some of those features in Table 3.
- *BM25:* We use the BM25 similarity measure trained on the cQA raw data provided by (Màrquez et al., 2015).
- *IR:* This is the order of the related questions provided by the search engine for question retrieval task and is the chronological ranking, in which answers are ordered by their time of posting, for the answer selection task.
- *TF-IDF:* This is computed using the cQA raw data provided by (Màrquez et al., 2015), and the ranking is defined by the cosine similarity of the TF-IDF vectors for the questions and answers.

We evaluate our models using F1-score for a global assessment of the models in addition to the following ranking metrics: Mean Average Precision (MAP), Average Recall (AveRec) and Mean Reciprocal Rank (MRR). For the MAP, we use the average of MAP@1 to MAP@10.

Method	Dev					
	MAP	AveRec	MRR	**F1**	R	P
BOV	63.18	82.56	69.36	56.84	52.08	62.56
BM25	55.16	73.18	63.33	-	-	-
IR	53.84	72.78	63.13	-	-	-
TF-IDF	52.52	72.34	60.20	-	-	-
Single LSTM - F_{aug}	61.25	81.76	68.57	-	-	-
Single BLSTM - F_{aug}	62.51	82.35	69.61	51.69	42.91	**65.00**
Single BLSTM	65.46	85.22	72.78	**62.47**	**63.69**	61.29
Double BLSTMs	**66.27**	**85.52**	**73.33**	60.36	59.66	61.08

(a) Results on development data for answer selection.

Method	Test					
	MAP	AveRec	MRR	**F1**	R	P
BOV	75.06	85.76	82.14	59.21	50.56	**71.41**
BM25	59.57	72.57	67.06	-	-	-
IR	59.53	72.60	67.83	-	-	-
TF-IDF	59.65	72.06	66.62	-	-	-
Single LSTM - F_{aug}	71.55	83.54	79.00	-	-	-
Single BLSTM - F_{aug}	73.29	84.58	80.82	53.00	42.89	69.34
Single BLSTM	74.03	85.49	82.53	62.91	59.67	66.53
Double BLSTMs	74.98	**85.98**	**83.05**	**63.53**	**59.89**	67.63

(b) Results on test data for answer selection.

Table 4: Results on (a) development and (b) test data for answer selection task in cQA.

Performance for Answer selection: The results of the answer selection task on development and test data are respectively shown in Tables 4a and 4b. In the tables, the first four rows show the baseline results, and the following rows show the neural models results. The "*Single LSTM - F_{aug}*" row shows the results of the model presented by Mohtarami et al. (2016) when only one LSTM is used instead of two bidirectional LSTMs, and no augmented features F_{aug} are used. The "*Single BLSTM - F_{aug}*" row indicates the results when one bidirectional LSTM is used in our model, and no augmented features F_{aug} are used. Using a BLSTM improves the performance compared to the single LSTM, as can be seen in Tables 4a and 4b. The "*Single BLSTM*" row shows the results for one bidirectional LSTM using F_{aug}. F_{aug} is a 10-length binary vector that encodes the order of the answers in their threads corresponding to their time of posting. F_{aug} helps improve the performance, as can be seen by comparing the results with the ones obtained using a single BLSTM without F_{aug}. The "*Double BLSTM*" row shows the results generated by the complete model illustrated in Figure 2. For the development set represented in Table 4a, the highest results over all the evaluation metrics are obtained using the neural models. The "*Double BLSTM*" achieves the highest performance over the ranking metrics. In addition, the results on the test set shown in Table 4b indicate that while the MAPs of the "*Double BLSTM*" and BOV baseline are comparable, the "*Double BLSTM*" achieves the highest performance over the other metrics, especially F1.

Performance for Question Retrieval: The results of question retrieval task on development and test data are respectively shown in Tables 5a and 5b. In the tables, the first four rows show the baseline results, and the following rows show the neural models results. The neural models are the ones described in the previous section. In this task, we employ the order of the related questions, provided by the search engine, as augmented features F_{aug} explained under *IR baseline* in Section 4. As shown in the tables, the neural models using F_{aug} outperform the models without F_{aug} for both development and test data. For the development set shown in Table 5a, the "*Double BLSTM*" model achieves the highest performance over the evaluation metrics. For the test set shown in Table 5b, the result of the "*Single BLSTM*" model is comparable with the IR and TF-IDF over the ranking metrics, while the highest F1 is obtained using BOV baseline. There are several points to highlight regarding the performance of the neural models compared to the baselines, : First, the size of the data for this task is small, which makes it harder to train our neural models. Second, the baselines have access to external resources; for example IR had ac-

Method	Dev					
	MAP	AveRec	MRR	**F1**	R	P
BOV	64.60	80.83	71.42	59.55	49.53	**74.65**
BM25	61.31	79.42	69.27	-	-	-
IR	71.35	86.11	76.67	-	-	-
TF-IDF	63.40	81.74	70.43	-	-	-
Single LSTM - F_{aug}	54.49	73.39	62.00	-	-	-
Single BLSTM - F_{aug}	57.00	74.54	62.85	51.64	51.40	51.89
Single BLSTM	67.40	83.14	75.87	44.94	37.38	56.34
Double BLSTMs	**70.75**	**86.2**	**76.83**	**62.83**	**66.36**	59.66

(a) Results on development data for question retrieval.

Method	Test					
	MAP	AveRec	MRR	**F1**	R	P
BOV	66.27	82.40	77.96	**56.81**	51.93	**62.69**
BM25	67.27	83.41	79.12	-	-	-
IR	74.75	88.30	83.79	-	-	-
TF-IDF	73.95	87.50	84.55	-	-	-
Single LSTM - F_{aug}	45.24	67.12	52.07	-	-	-
Single BLSTM - F_{aug}	48.00	70.39	54.18	40.88	48.07	35.56
Single BLSTM	73.20	86.99	83.38	48.15	44.64	52.26
Double BLSTMs	71.98	85.86	81.16	51.27	**64.81**	42.42

(b) Results on test data for question retrieval.

Table 5: Results on (a) development and (b) test data for question retrieval task in cQA.

cess to the click log of the users and TF-IDF is trained on a large cQA raw dataset (Màrquez et al., 2015). Finally, the number of out-of-vocabulary (OOV) words in the test data is higher than the development data, and the OOV word vectors are randomly initialized and do not get updated during the training phase. This results in a smaller improvement on the test data.

4.1 Visualization

In order to gain better intuition on our neural model, we consider our complete model with two bidirectional LSTMs as illustrated in Figure 2, and represent the outputs of the hidden layers for each bidirectional LSTM. The represented outputs correspond to the cosine similarities between word vector representations of words in question-question pairs or question-answer pairs. Figure 3 shows the heatmaps for the first bidirectional LSTM (top) and the second bidirectional LSTM (bottom) for the question retrieval task with the following two questions:

- q_1: Which is the best Pakistani school for children in Qatar ? Which is the best Pakistani school for children in Qatar ?
- q_2: Which Indian school is better for the kids ? I wish to admit my kid to one of the Indian schools in Qatar Which is better DPS or Santhinekethan ? please post your comments

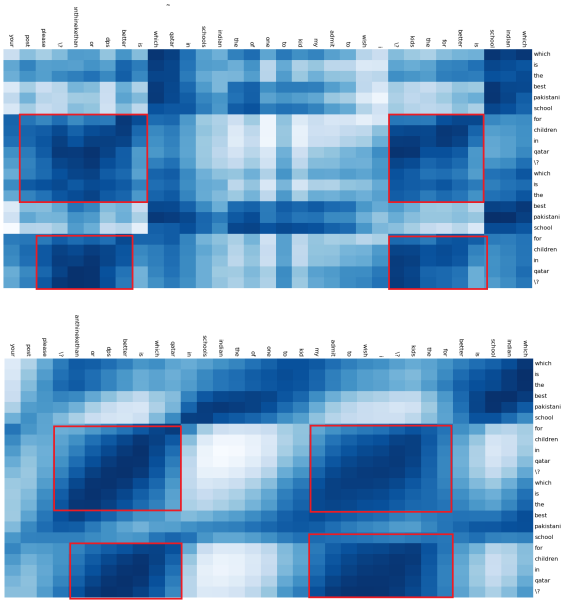

Figure 3: Example of a pair of questions that is correctly predicted as similar by the first (top) and second (bottom) bidirectional LSTMs. The dark blue squares represent areas of high similarity.

The areas of high similarity are highlighted in the red squares in figure 3. While both bidirectional LSTMs correctly predict that the questions are similar, the heatmaps show that the second bidirectional LSTM performs better than the first one, and that the areas of similarities (delimited by the red rectangles) are much better defined by the second bidirectional LSTM. For ex-

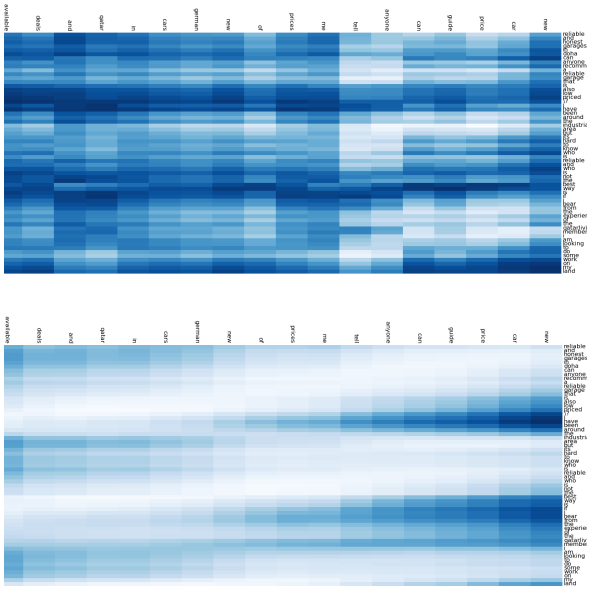

Figure 4: Example of a pair of questions that is incorrectly predicted as similar by the first bidirectional LSTM (top) and correctly predicted by the the second bidirectional LSTM (bottom). The dark blue squares represent areas of high similarity.

ample, the first bidirectional LSTM identifies similarities between the part "*for children in qatar ? Which is the*" from the question q_1 with the parts "*is better for the kids ?*" and "*is better DPS or Santhinekethan ?* ***please post***" from the question q_2. The second bidirectional LSTM accurately updates those parts from the question q_2 to "*for the kids ? I wish to* ***admit*** *my*" and "*Qatar which is better DPS or Santhinekethan*" respectively. This shows that the second bidirectional LSTM assigns smaller values to the non-important words (e.g., "*please post*") while highlighting important words (e.g., "*admit*").

Figure 4 shows the heatmaps for the first bidirectional LSTM (top) and the second bidirectional LSTM (bottom) for another example of the question retrieval task with the following two questions:

- q_3: New car price guide. Can anyone tell me prices of new German cars in Qatar and deals available

- q_4: Reliable and honest garages in Doha. Can anyone recommend a reliable garage that is also low priced ? I have been around the industrial area but it is hard to know who is reliable and who is not. The best way is if I hear from the experience of the qatarliving members . I am looking to do some work on my land cruiser

As shown in the figure, the areas highlighted in dark blue in the first bidirectional LSTM are much larger than the second bidirectional LSTM. These results show that the first bidirectional LSTM incorrectly predicts that the questions q_3 and q_4 are similar, while the second bidirectional LSTM correctly predicts that the questions are dissimilar.

5 Conclusion

In this paper, we present a neural-based model with stacked bidirectional LSTMs to generate the vector representations of questions and answers, and predict their semantic similarities. These similarity scores are then employed to rank elements in a list of questions in the question retrieval task, and a list of answers in the answer selection task for a given question. The experimental results show that our model can perform better than the baselines, even though the baselines use various text- and vector-based features and have access to external resources. We also demonstrate the impact of the OOV words, and the size of the train data on the performance of the neural model.

Acknowledgments

This work is supported by the Qatar Computing Research Institute (QCRI). We thank members of the MIT Spoken Language Systems (SLS) group and the reviewers for their helpful comments.

References

Michael Auli, Michel Galley, Chris Quirk, and Geoffrey Zweig. 2013. Joint language and translation modeling with recurrent neural networks. In *EMNLP*, volume 3.

Alberto Barrón-Cedeno, Simone Filice, Giovanni Da San Martino, Shafiq Joty, Lluıs Marquez, Preslav Nakov, and Alessandro Moschitti. 2015. Threadlevel 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, ACLIJCNLP*, volume 15, pages 687–693.

Yonatan Belinkov, Mitra Mohtarami, Scott Cyphers, and James Glass. 2015. Vectorslu: A continuous word vector approach to answer selection in community question answering systems. *SemEval-2015*, page 282.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5(2):157–166.

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*, pages 192–199. ACM.

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

Robin D Burke, Kristian J Hammond, Vladimir Kulyukin, Steven L Lytinen, Noriko Tomuro, and Scott Schoenberg. 1997. Question answering from frequently asked question files: Experiences with the faq finder system. *AI magazine*, 18(2):57.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

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)*, pages 694–699, Beijing, China, July. Association for Computational Linguistics.

Christiane Fellbaum. 1998. A semantic network of english verbs. *WordNet: An electronic lexical database*, 3:153–178.

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. *CoRR*, abs/1508.01585.

Alex Graves and Navdeep Jaitly. 2014. Towards end-to-end speech recognition with recurrent neural networks. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1764–1772.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610.

Jakob Grundström and Pierre Nugues. 2014. Using syntactic features in answer reranking. In *AAAI 2014 Workshop on Cognitive Computing for Augmented Human Intelligence*, pages 13–19.

Michael Heilman and Noah A Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1011–1019. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Yongshuai Hou, Cong Tan, Xiaolong Wang, Yaoyun Zhang, Jun Xu, and Qingcai Chen. 2015. Hitszicrc: Exploiting classification approach for answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval*, volume 15, pages 196–202.

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*, pages 84–90. ACM.

Jiwoon Jeon, W Bruce Croft, Joon Ho Lee, and Soyeon Park. 2006. A framework to predict the quality of answers with non-textual features. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 228–235. ACM.

Shafiq Joty, Alberto Barrón-Cedeno, Giovanni Da San Martino, Simone Filice, Lluıs Marquez, Alessandro Moschitti, and Preslav Nakov. 2015. Global thread-level inference for comment classification in community question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP*, volume 15.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137.

Harksoo Kim and Jungyun Seo. 2006. High-performance faq retrieval using an automatic clustering method of query logs. *Information processing & management*, 42(3):650–661.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Jan Koutnik, Klaus Greff, Faustino Gomez, and Juergen Schmidhuber. 2014. A clockwork rnn. *arXiv preprint arXiv:1402.3511*.

Yu-Sheng Lai, Kuao-Ann Fung, and Chung-Hsien Wu. 2002. Faq mining via list detection. In *proceedings of the 2002 conference on multilingual summarization and question answering-Volume 19*, pages 1–7. Association for Computational Linguistics.

Shujie Liu, Nan Yang, Mu Li, and Ming Zhou. 2014. A recursive recurrent neural network for statistical machine translation. In *ACL (1)*, pages 1491–1500.

Bernardo Magnini, Matteo Negri, Roberto Prevete, and Hristo Tanev. 2002. Is it the right answer?: exploiting web redundancy for answer validation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 425–432. Association for Computational Linguistics.

Christopher D Manning and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*, volume 999. MIT Press.

Lluís Màrquez, James Glass, Walid Magdy, Alessandro Moschitti, Preslav Nakov, 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 2015)*.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH*, volume 2, page 3.

Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Honza Černockỳ, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5528–5531. IEEE.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mitra Mohtarami, Yonatan Belinkov, Wei-Ning Hsu, Yu Zhang, Tao Lei, Kfir Bar, Scott Cyphers, and James Glass. 2016. Sls at semeval-2016 task 3: Neural-based approaches for ranking in community question answering. In *Proceedings of NAACL-HLT Workshop on Semantic Evaluation*, pages 753–760, San Diego, California, June. Association for Computational Linguistics.

Alessandro Moschitti, Silvia Quarteroni, Roberto Basili, and Suresh Manandhar. 2007. Exploiting syntactic and shallow semantic kernels for question answer classification. In *Annual meeting-association for computational linguistics*, volume 45, page 776.

Alessandro Moschitti. 2008. Kernel methods, syntax and semantics for relational text categorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 253–262. ACM.

Preslav Nakov, Lluís Màrquez, Walid Magdy, Alessandro Moschitti, 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, June. Association for Computational Linguistics.

Massimo Nicosia, Simone Filice, Alberto Barrón-Cedeno, Iman Saleh, Hamdy Mubarak, Wei Gao, Preslav Nakov, Giovanni Da San Martino, Alessandro Moschitti, Kareem Darwish, et al. 2015. Qcri: Answer selection for community question answeringexperiments for arabic and english. In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval*, volume 15, pages 203–209.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2012. On the difficulty of training recurrent neural networks. *arXiv preprint arXiv:1211.5063*.

Filip Radlinski and Thorsten Joachims. 2005. Query chains: learning to rank from implicit feedback. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 239–248. ACM.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on*, 45(11):2673–2681.

Aliaksei Severyn and Alessandro Moschitti. 2012. Structural relationships for large-scale learning of answer re-ranking. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 741–750. ACM.

Aliaksei Severyn and Alessandro Moschitti. 2013. Automatic feature engineering for answer selection and extraction. In *EMNLP*, pages 458–467.

Aliaksei Severyn and Alessandro Moschitti. 2015a. 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*, pages 373–382. ACM.

Aliaksei Severyn and Alessandro Moschitti. 2015b. 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, New York, NY, USA. ACM.

Dan Shen and Mirella Lapata. 2007. Using semantic roles to improve question answering. In *EMNLP-CoNLL*, pages 12–21.

Eriks Sneiders. 2002. Automated question answering using question templates that cover the conceptual model of the database. In *Natural Language Processing and Information Systems*, pages 235–239. Springer.

Wanpeng Song, Min Feng, Naijie Gu, and Liu Wenyin. 2007. Question similarity calculation for faq answering. In *Semantics, Knowledge and Grid, Third International Conference on*, pages 298–301. IEEE.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2008. Learning to rank answers on large online qa collections. In *ACL*, volume 8, pages 719–727.

Ilya Sutskever, James Martens, and Geoffrey E Hinton. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1017–1024.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ming Tan, Bing Xiang, and Bowen Zhou. 2015. Lstm-based deep learning models for non-factoid answer selection. *CoRR*, abs/1511.04108.

Kateryna Tymoshenko and Alessandro Moschitti. 2015. Assessing the impact of syntactic and semantic structures for answer passages reranking. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1451–1460. ACM.

Mengqiu Wang and Christopher D Manning. 2010. Probabilistic tree-edit models with structured latent variables for textual entailment and question answering. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1164–1172. Association for Computational Linguistics.

Mengqiu Wang, Noah A Smith, and Teruko Mitamura. 2007. What is the jeopardy model? a quasi-synchronous grammar for qa. In *EMNLP-CoNLL*, volume 7, pages 22–32.

Steven D Whitehead. 1995. Auto-faq: An experiment in cyberspace leveraging. *Computer Networks and ISDN Systems*, 28(1):137–146.

Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013. Answer extraction as sequence tagging with tree edit distance. In *HLT-NAACL*, pages 858–867. Citeseer.

Kaisheng Yao, Trevor Cohn, Katerina Vylomova, Kevin Duh, and Chris Dyer. 2015. Depth-gated recurrent neural networks. *arXiv preprint arXiv:1508.03790*.